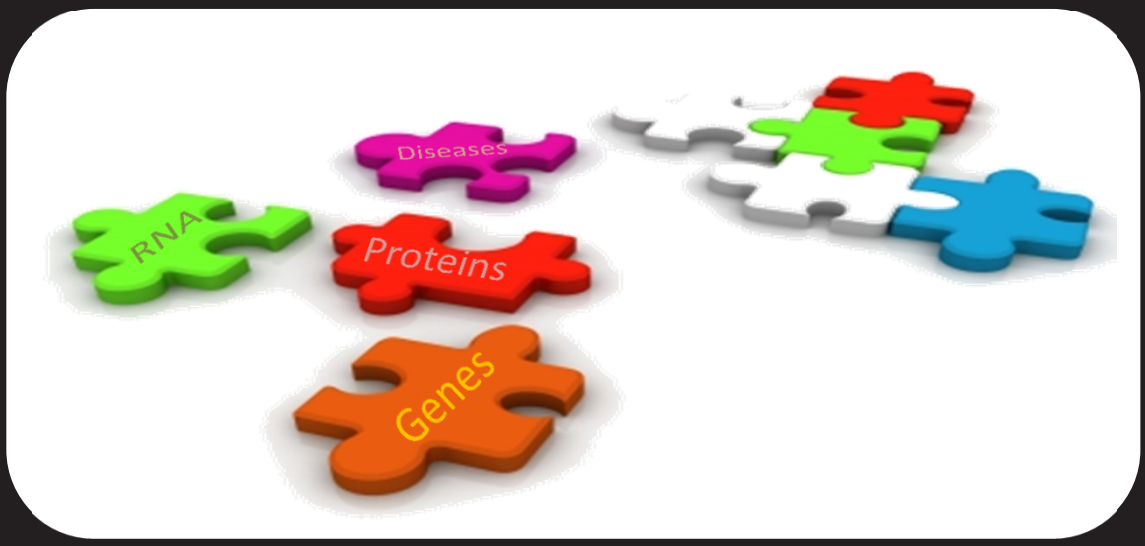
Improving discovery in the Life Sciences using Semantic Web Technologies and Linked Data

Design principles for Life Sciences Knowledge Organization Systems

Helena Futscher de Deus



Dissertation presented to obtain the Ph.D degree in Bioinformatics

Instituto de Tecnologia Química e Biológica | Universidade Nova de Lisboa

Oeiras,
April, 2011

INSTITUTO
DE TECNOLOGIA
QUÍMICA E BIOLÓGICA
/UNL

Knowledge Creation

Improving discovery in the Life Sciences using Semantic Web Technologies and Linked Data

Design principles for Life Sciences Knowledge Organization Systems

Helena Futscher de Deus

Dissertation presented to obtain the Ph.D degree in Bioinformatics

Instituto de Tecnologia Química e Biológica | Universidade Nova de Lisboa

Research work coordinated by:

FUNDAÇÃO CALOUSTE GULBENKIAN
Instituto Gulbenkian de Ciência

Work funded by:

FCT
Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

Cover Image: Illustration of the recurrent need, in biological discovery, to put together the pieces of the systemic puzzle to fully understand biology and its effect on disease.
Attribution: "3d Puzzle Pieces" by jscreationzs

Oeiras,
April, 2011

INSTITUTO
DE TECNOLOGIA
QUÍMICA E BIOLÓGICA
/UNL

Knowledge Creation

Acknowledgements

First and foremost, I dedicate this thesis to my family, for always being there. A very warm Thank You to my mother for her unconditional love and all the support that she has given and continues to give. I thank my father for always helping me look beyond the horizon. To my little sister, Teresa, I'm sure we'll be seeing you some day producing great animated movies (and Thanks for all the animation tips!)

To all my Houston friends and colleagues, for making life there a lot more colorful… To all my sailing buddies, proud followers of the religion of the Holy Spinnaker, for all their words of encouragement during my journey! And to the high-priest and high-priestess themselves, John and Helen, I love you both! Thank you for sharing with me your passion for sailing.

To my PhD colleagues at ITQB.

My Thank You goes also to Professora Hermínia de Lencastre, who believed in me even when I was still a recent graduate in Marine Biology with high hopes. My thank you to both the Molecular Genetics Lab and the former Biomathematics "gang" at ITQB, your help, support and patience for teaching me both biology and programming helped me climb mountains I never thought I could…

Last but not least, to my PhD Advisor Jonas S. Almeida who gave me all I ever wanted during my PhD and, more than that, for giving me all that I ever *needed* during the 5 years we have worked together (even when I didn't realize that I needed it at the time)! Jonas helped me to grow not only as a scientist but as a person, for that I am eternally grateful. Thank you!

The work presented in this thesis was financially supported by the Center for Clinical and Translational Sciences under contract no. 1UL1RR024148 and by the FCT scholarship SFRH/BD/45963/2008.

Table of Contents

Thesis statement

One of the most enticing outcomes of biological exploration for the quantitative-minded researcher is to identify the “mathematics of biology” through the study of the patterns and interrelatedness of biological entities. To move forward in that direction, many pieces need to be set in place, from the availability of biological data in forms that can be computationally manipulated to the automated discovery of patterns that would derive from integration of data from many and diverse endpoints. A computational system where researchers could securely deposit any type of data and have it immediately analyzed, traversed, annotated and merged with data deposited elsewhere is a dream not yet achieved but one which could revolutionize scientific discovery. The set of technologies that would make such a system possible were for the first time proposed 10 years ago and have since been maturing to the point where they have become useful for actual deployment in Life Sciences applications. This set of technologies, termed Semantic Web technologies, were proposed with the goal of transforming the World Wide Web into a computational platform, where both read and write operations are supported and the computational boundaries that exist between knowledge systems are erased.

The work presented in this thesis is a stepping stone in making such systems possible. A prototypical tool named S3DB (Simple Sloppy Semantic Database) was initially developed following the principles set forth by the Semantic Web community. The S3DB prototype was evolved and validated following the need of several Life Sciences groups with interests that ranged from molecular epidemiology, to multiple ‘omics’ data derived from a multibillion cancer project, to the methodologies involved in clinical trial deployment. The minimum set of generic principles required to support integrated knowledge management of distributed biomedical applications came as an iterative process from multiple interactions with the users of the tool.

Outline

In the first chapter, the challenges and requirements for knowledge engineering in the Life Sciences are presented. The two approaches typically used when devising the representation systems that support knowledge engineering efforts are introduced and its limitations explained. We then present the basic concepts underlying our solution, the S3DB knowledge organization system, and discuss the key requirements that should be met when applying semantic web technologies to Life Sciences knowledge engineering.

The second chapter compiles the foundational work, described in three published articles, that illustrate the key technological requirements for supporting knowledge representation for the Life Sciences: “Data Integration gets ‘Sloppy’” discusses the need for flexible data models that can be edited by the data expert to justify the S3DB architecture; “A Semantic Web Management Model for Integrative Biomedical Informatics” illustrates the application of semantic web technologies to biomedical data management with resort to S3DB; and finally in “S3DB core: a framework for RDF generation and management in bioinformatics infrastructures” the generic S3DB framework is formally described.

In the third chapter, we apply the principles presented in the second chapter to two Life Sciences applications. In “Exposing the cancer genome atlas as a SPARQL endpoint”, the large interdisciplinary and multi-institutional dataset from The Cancer Genome Atlas (TCGA) project is made available to semantic web technologies through a SPARQL endpoint. This was achieved through mediation by the S3DB prototype. The technical advance presented in “AGUIA: autonomous graphical user interface assembly for clinical trials semantic data services” makes use of the S3DB prototype for the automated assembly of interfaces that are intuitive to clinical trial research experts.

The fourth chapter corresponds to “S3QL: A distributed domain specific language for controlled semantic integration of Life Sciences data”, where we propose a solution to the creation of “social machines” by controlled read/write operations in entities of the S3DB Core Model. Although S3QL corresponds to a component of the S3DB system and would likely be a suitable candidate for the second chapter, its development was driven and validated by the needs of the applications presented in the third chapter. As such, we consider that the S3QL protocol described here is, in many ways, the corollary of the work performed and is also probably its most lasting deliverable.

Summary (English)

The data deluge in biology resulting from wide adoption of high-throughput technologies, coupled with the increasing reliance on web technologies for knowledge organization, sharing and discovery, has created unprecedented opportunities, and challenges, for knowledge engineering in Life Sciences domains. The Semantic Web technologies correspond to a set of standards and best practices for improving data sharing and interoperability on the Web that can greatly advance research in data-driven sciences such as translational medicine and systems biology. Current Semantic Web approaches for addressing those challenges have either relied on automatically formatting biological data sources as RDF (Resource Description Framework), the lingua franca of the Semantic Web, or in the development of bio-ontologies. Albeit the significant integrative advances that those represent, wide adoption of Semantic Web technologies by the communities acquiring and modeling experimental biological data has remained suboptimal.

We present here a novel approach to Semantic Web knowledge representation in biomedical domains that relies on empowering biomedical domain experts with the tools to weave the representation models that best suit their data management, visualization and integration needs. With that goal, we have developed the S3DB (Simple Sloppy Semantic Database) model, a Knowledge Organization System for the Life Sciences, and applied its core principles in a prototypical application. Two key requirements were identified in the process which correspond to the need for flexible, distributed data representation models that can be collaboratively evolved to include additional experimental parameters and a fine grained permission management system. We have correspondingly addressed those requirements in the S3DB model with two technical advances: the separation of the data descriptors from the data elements through identification of a distributed management model for S3DB and the

description of a set of operators, with states, that propagate according to a set of formally defined equations (s3db:operators).

To validate our approach, the S3DB prototype was applied in three biomedical use cases from diverse knowledge domains including Clinical Trials, Molecular Epidemiology and Cancer. A critical component for automating data management and visualization, as well as integration with external datasets, was the identification of the domain specific language S3QL. A set of management actions controlled by the *s3db:operators* were defined for S3QL to enable read/write operations on entities of the S3DB model. We illustrate how S3QL can be easily adopted as a programming interface for knowledge organization systems beyond S3DB. The contextual representation supported by our model is set to become increasingly significant as the tools to share and manage biomedical data move closer to online social and collaborative ecosystems.

Sumário (Português)

O dilúvio de dados em Biologia resultante de adopção das tecnologias de alto rendimento, acoplado à cresceste utilização de tecnologias da Web para organizar, partilhar e descobrir novo conhecimento nos domínios das Ciências Biológicas, criou oportunidades e desafios sem precedentes para as engenharias do conhecimento aplicadas aos domínios Biológicos. As tecnologias da Web Semântica, que correspondem a um conjunto de padrões e recomendações para melhorar a partilha e interoperabilidade de dados na Web, apresentam um enorme potencial para o avanço da pesquisa baseada nos dados tais como a investigação de translação e a biologia de sistemas. Actualmente, as abordagens para a aplicação da Web Semântica na resolução destes desafios têm-se focado ora na formatação automática de fontes de dados Biológicos em RDF (Resource Description Framework), a língua franca da Web Semântica, ora no desenvolvimento de bio-ontologias. Embora a aplicação destas abordagens tenha resultado em avanços significativos na integração de dados, a adopção em larga escala das tecnologias da Web Semântica pelas comunidades que adquirem e modelam dados biológicos experimentais tem permanecido a um nível sub-óptimo.

Este trabalho apresenta uma nova abordagem para a representação de conhecimento em domínios biomédicos na Web Semântica que se baseia na criação de ferramentas que permitem aos investigadores especializados em biomedicina criar os modelos de representação que mais se adequam às suas necessidades de gestão, visualização e integração dos dados. Com esse objectivo, foi desenvolvido o modelo S3DB (Simple Sloppy Semantic Database), um sistema de organização do conhecimento para as Ciências Biológicas, e os seus princípios nucleares foram aplicados na criação de uma aplicação protótipo. Neste processo foram identificados dois requisitos fundamentais que correspondem à necessidade de modelos de representação flexíveis e distribuídos que possam ser colaborativamente

editados de forma a incluir parâmetros experimentais adicionais, bem como um sistema de gestão de permissões de alta granularidade. Estas necessidades foram supridas no modelo S3DB através de dois avanços tecnológicos: a separação entre a descrição e os elementos dos dados, através da identificação de um modelo de gestão distribuído para a S3DB, e da descrição de um conjunto de operadores com estados que são propagados de acordo com equações formalmente definidas (*s3db:operators*).

De forma a validar a abordagem desenvolvida, o protótipo S3DB foi aplicado a três casos de estudo biomédicos em diversos domínios do conhecimento incluindo Ensaios Clínicos, Epidemiologia Molecular e Cancro. Um componente que se revelou crítico para a automatização da visualização e gestão de dados, bem como para a sua integração com conjuntos de dados externos, foi a identificação de uma linguagem de domínio específico denominada S3QL. Foi definido um conjunto de acções de gestão controladas através de s3db:operators para a S3QL, de forma a permitir ler e escrever em entidades do modelo S3DB. Demonstra-se que a S3QL pode facilmente ser adaptada como uma interface de programação para sistemas de representação do conhecimento alternativos ao S3DB. A representação contextual suportada por este modelo poderá vir a tornar-se uma vantagem significativa à medida que as ferramentas para partilha e gestão de dados biomédicos se aproximam de ecossistemas sociais de colaboração em rede.

Glossary

Abox

The assertional component of a knowledgebase

Access Protocols

A set of protocols, usually computational, used to exchange information between two endpoints, which can be two computers or two persons.

Data Model

A user-defined view of data representing the world, usually including the details of various subjects and their properties

Database Surfing

The act of browsing online database interfaces while searching for different pieces of relevant information to answer a question

Domain Expert

A person, usually non-IT expert, with special knowledge on an area of research

Fine Grained Permission Control

A model for permission control that enables definition of permissions at the level of individual data elements

Functional Genomics

A branch of genomics that studies the biological function of the genes and their products

Gbox

The graphical display component of a knowledgebase

High-Throughput Assays

Automated processes of screening for thousands of substances, including genes and proteins

HTTP

Hypertext Transfer (or Transport) Protocol, the data transfer protocol used on the World Wide Web

Inference

The process of creating assertions based on other assertions

Interoperability

A property of a computational framework that relies on the ability of diverse computational systems, hosting different types of data and working with different protocols, to effectively exchange and reuse information.

Knowledgebase

A database designed to meet the storage and retrieval requirements of computational knowledge representation. It typically includes an assertional (Abox) and a terminological component (Tbox), although it may also include a graphical description component (Gbox).

Knowledge Engineering

An engineering discipline that involves integrating knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise

Knowledge Organization System (KOS)

A scheme for organizing information and promoting knowledge management. Examples include glossaries, subject headings, taxonomies and ontologies.

'-omic' Sciences

The study of biological systems using high-throughput assays. Examples include proteomics, genomics, and metabolomics.

Ontologies

A formal representation of knowledge by a set of concepts within a domain and the relationships between those concepts, usually with a hierarchical organization.

Orthogonal

A property whereby each variable or concept occurs only once. The term is applied in the context of knowledge representation systems to refer to ontologies with non-overlapping domains.

OWL (Web Ontology Language)

A vocabulary extension of the Resource Description Framework (RDF), representing the logic relationships between terms and vocabularies in such a way that it can be used to inference new assertions.

Query Federation

The interoperation of distinct, formally disconnected, knowledgebases to assemble information stored across multiple systems, joined together by shared common elements such as URI.

RDF (Resource Description Framework)

A specification for describing relationships between resources through triples composed a subject, a predicate and an object. Each element in the triples is identified by a URI.

RDFS (RDF Schema)

A vocabulary extension of the Resource Description Framework (RDF) that includes the logic formalisms for describing groups of related resources and their relationships.

REST (Representational State Transfer)

An HTTP based protocol that can be used to request a service through a web browser via appropriately formatted URL of by posting HTML forms.

Semantic Web

The third generation of web technologies that enables data on the web to be made reusable through the creation of machine-readable information.

SparQL

A recursive acronym that describes the SPARQL Protocol And RDF Query Language. It is considered to be the query language for the Semantic Web.

Systems Biology

The systematic study of complex interactions in biological systems by applying mathematical rules and principles that describe and predict biological behaviors.

Tbox

The terminology component of a knowledgebase.

Translational Research

The translation of biological discoveries derived from basic research into products that can be used in clinical trials and eventually into clinical intervention.

URI (Universal Resource Identifier)

A string of characters used to identify a name or a resource on the Web. A URI may correspond to a URL (Universal Resource Locator), but may also result from the concatenation of a URL, the “hash” signal and a string identifying a resource within that URL

XML (eXtended Markup Language)

A meta-language, composed of a set of rules for encoding documents electronically, that can be used to define customized markup languages.

Chapter 1 | Introduction

"Science is organized knowledge", said Immanuel Kant [1]. Science in the Information Age has indeed been characterized by increasingly complex approaches to the organization and sharing of scientific knowledge that address its increasingly systemic nature and improve its discovery [2,3]. Knowledge engineering (KE) is the sub-discipline of artificial intelligence concerned with the automation of knowledge intensive tasks, such as problem solving methods, which would otherwise require the aid of a human expert [4]. This is achieved by relying on models that computationally represent knowledge domains. In biomedical research, KE has been intensively used in computational medical decision support systems to aid health care staff make informed decisions regarding a patient's diagnosis and treatment [5]. More recently, research in biomedical KE has been driven by a need to support the translation of knowledge acquired through molecular biology research into clinical practice [6-8]. Investment in translational research has indeed been considered a priority both by the National Institutes of Health of the United States [9,10] and by European Commission FP7 Health research program [11].

A critical bottleneck in applying knowledge engineering to inform and improve biomedical discovery is the ability to align data acquisition efforts with knowledge representation models [12]. Recent technological advances in genomics, proteomics and other 'omics' sciences, have flooded microbiology labs with unprecedented amounts of data and were at the heart of a paradigm shift for researchers interested in its computational representation and analysis [13-15]. Data-driven biomedical researchers quickly realized that these high-throughput techniques could be used for simultaneously measuring multiple biological elements and thus enable systematic approaches to the study of disease mechanisms and discovery of functional genomics associations [16]. Computational systems biology, for

example, is a data-driven discipline that makes use of this data deluge for creating mathematical and computational models aimed at explaining, and often predicting, the biological mechanisms underlying disease [17-19]. In order to understand a biological system's structure and dynamics, and to computationally model it, access to a continuum spectrum of domain data and knowledge is essential [7,20,21]. Achieving the goals set forth by systems biology and translational research therefore requires the seamless integration of vast amounts of biological data from multiple knowledge domains and with multiple orders of magnitude. This represents a challenge for Life Sciences KE, not only due to the heterogeneous and diverse nature of the biological domains with their constant pursuit for new measurable parameters, which lead to constant revision of the data representation models, but also because such advances must be pursued without compromising the privacy of patient data [22-24].

The Role of Integrative Bioinformatics in creating a Knowledge Continuum

The deluge of data in biology marked the beginning of the "industrialization of data production in Life Sciences beyond a craft-based cottage industry" [25] and predisposed the field of integrative bioinformatics to following the same route. The successful exploit of the new wave of data acquisition technologies created the need for agreement on the optimal strategies for sharing and modeling biomedical data that enabled experimental replication of results [15,26,27]. In microarray technologies, for example, large scale community efforts resulted in the definition of the minimal information about a microarray experiment (MIAME) standard, a document model specifying the minimum set of experimental variables necessary to replicate a microarray experiment [28]. Such standards were welcomed by their respective communities and helped shape the knowledge representation models that computationally represent its data. Microarray experiment databases such as GEO [29] or ArrayExpress [30] have since included MIAME standards in their database schemas.

The increasing complexity of Life Sciences datasets also stimulated the emergence of specialized communities dedicated to increasing the value and usefulness of data by interlinking it with related data. Such efforts have resulted in highly successful technologies such as Ensembl [31], the UCSC genome browser [32], BLAST [33], Uniprot [34] or Gene Ontology [35], which support reuse and integration of experimental results with knowledge generated elsewhere. These technologies have been adopted by a variety of research communities and are today considered critical enablers of scientific discovery [36]. Such technological achievements, however, rely on a large amount of community effort and involvement, making it a time-consuming process that requires extracting, transforming and remodeling data from

multiples sources [37,38]. Moreover, each individual platform was devised to suit the knowledge representation needs of the biology domain experts that maintain them and address problems confined within their respective domain boundaries. This creates a conundrum for biomedical researchers because the answer to a biological problem often spans across multiple domain boundaries. As a result, many reasonable biological questions require biomedical researchers to become experts at “database surfing” [36].

Multiple attempts have been made to enable straightforward traversal of multiple biological data sources. The most common strategies involve either data warehousing or view-link integration [36,38,39]. The former relies on the identification of a unified model for aggregating data from multiple sources. In the latter, the data necessary to answer a query is dynamically identified by relying on linking formalisms agreed upon by the communities using and maintaining the data sources. Nonetheless, the issue of integrative bioinformatics has remained problematic, which suggests that substantial underlying issues remain unresolved [25]. On the one hand, although most biological data sources are architecturally similar, there are profound technical differences both in the access protocols and incompatible data models as each has been optimized to answer a specific class of problems [36]. Furthermore, constant updates of the data models defy automated, persistent integration. On the other hand, there are deep inconsistencies in the description of the data which further challenge integration: 1) the names and identifiers of biological entities are not easily mapped across data sources and 2) the concepts describing data elements often clash when transposed from one source to another. The protocols to programmatically obtain annotated experimental results from biological data sources have also not been completely standardized [40] with few exceptions such as, for example, the distributed annotation system [41].

Semantic Web Approaches to Knowledge Integration

The systems devised for integrating biological data sources have mostly relied on the technologies available at the time when they were first developed. Regardless of the technical details of each implementation, their intent is more or less the same: to easily and unambiguously traverse multiple data sources, regardless of the location or format of the data necessary to answer a biological question [42]. When relational technologies were available, methodologies were focused on developing mechanisms to mediate query and data translation [43]. When these could no longer support the flexibility needed to fully represent biological complexity, eXtended Markup Language (XML) schemas became the standard integration formalisms and a flurry of biological markup languages became available [44-48]. XML, it is argued, although more efficient in representing the hierarchical relationships between biological concepts, does not easily support extensions to the representation models that would be required to keep pace with rapidly evolving technologies [49]. The data-driven requirements of biomedical research are more successfully addressed by flexible representation formalisms [50-53] that support collaborative editions of the data models [54,55] and that scale well with the distributed nature of the World Wide Web, the most successful architecture to date for distributed data access and therefore the suitable platform to integrate disparate datasets. Efforts to develop such formalisms were driven by the knowledge engineering community and crystallized in the semantic web set of technologies and standards [56].

The Semantic Web (SW) was first proposed by the original architect of the World Wide Web as a global data space understandable not only by humans but also by machines [57]. The novelty in this proposal lies in two key principles that are meant to enable interoperability using finite effort and cost [58]: 1) HTTP universal resource identifiers (URI) should be used to identify data elements and 2) relationships between the data elements

should be formalized as “subject-predicate-object” triples [59]. For example, making use of SW technologies to explicitly represent a relationship between one entity of type “Gene” and one entity of type “Protein” is a two-step process: first, each entity is assigned a URI such as http://uniprot.org/protein (which may be referred to as uniprot:protein); second, a triple representation such as *[:gene :hasProduct uniprot:protein]* formalizes the relationship between the two. The World Wide Web Consortium (W3C), the authority responsible for maintaining Web protocols consistency, recommends that the Resource Description Framework (RDF) should be used when representing data in a triple format [60]. Although other triple representations exist, such as the Entity-Relationship-Entity model [61], RDF offers the advantage of relying on the widely supported HTTP protocol and therefore is more easily distributed over multiple web systems. The W3C has also proposed the formal standard for performing local and federated queries over individual RDF data sources using SPARQL, a protocol and RDF query language [62]. The set of best practices for making use of RDF and SPARQL in linking datasets and performing queries over multiple sources is descriptively called Linked Data [63]. The SW stack, the set of standard technologies that support the semantic web, also includes specification of the RDF Schema (RDFS) and the Web Ontology Language (OWL), standards for addressing automated logical inference [64-66].

Motivated by the potential that widespread application of SW technologies could have in improving scientific discovery and translational research, researchers have made use of the SW set of technologies both for domain knowledge representation and for data acquisition [12,67]. As a result, two mainstream approaches have dominated SW approaches when addressing the interdisciplinary requirements of Life Sciences knowledge engineering. These can be roughly classified as top-down and bottom-up approaches: top-down driven researchers see knowledge representation as

an end in itself, whereas bottom-up driven researchers see it as a means to an end which is the advancement of science [67] (Figure 1).

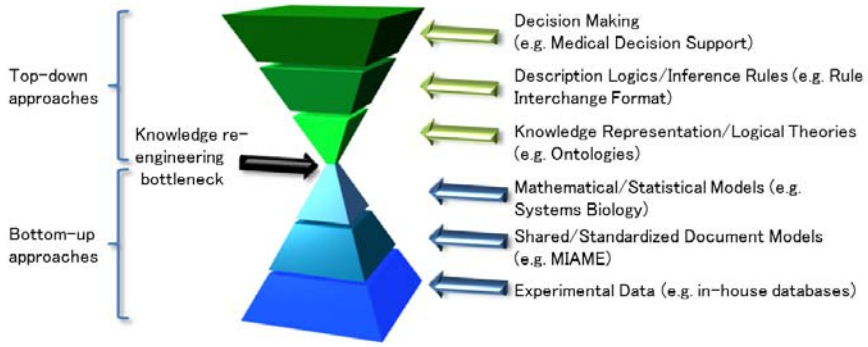


Figure 1. The Knowledge re-engineering bottleneck and the two approaches to knowledge engineering on either side – the top down and the bottom-up approaches – which tackle different aspects in opposite directions of the knowledge continuum.

Top-down approaches stem from the assumption that systemic integration requires domain communities to agree on ontologies, unified models of representation with embedded logic [68]. Ontologies provide conceptual representations of knowledge domains by describing the hierarchical relationships between concepts and logical rules that enable automated inference of assertions [69]. For example, using well engineered OWL formatted ontologies would enable asserting that a biological entity comprised of a single chain of 3 or more amino acids is of type "Polypeptide". The creation of Robot Scientists able to carry out "autonomous scientific discovery" [70] is also enabled by ontology engineering [71]. These achievements, however, require highly controlled knowledge representation models to ensure the orthogonality of the ontologies, i.e. that their domains do not overlap, and thus avoid inconsistencies such as contradictory assertions [72]. The opposing bottom-up approach is driven by the principle that scientific independence of biological data sources should be respected [25,73]. Researchers motivated

by this principle trust that, due to the burgeoning amounts of biological data and the continuous improvement of data models, global agreement will hardly be reached at the cutting edge of science, where knowledge representation is most needed [74,75]. Forced adherence to formal conceptual representation models in biomedical data management might indeed hamper the innovation and creativity that often drives scientific discovery [12,26,73]. Instead, widespread adoption of Linked Data principles is suggested, either by making widely used biomedical resources available as RDF [50,76-81] or by making use of tools such as Sesame [82] for representing new experimental datasets as RDF.

A compromise between the two views has been sought through the development of wiki-like portals [83,84] that empower biology domain experts to creating and collaboratively evolving the ontologies that best suit their data acquisition and management needs. To cross the unavoidable domain boundaries between representation models, domain and task independent ontologies such as, for example, the Simple Knowledge Organization System (SKOS) for modeling RDF data [85], have been developed without technological commitment to logical formalisms [66]. Additionally, bottom-up researchers make use of these ontologies as sources of controlled vocabularies [13]. Using basic semantic web formalisms, it is possible to address identifier and conceptual inconsistencies across datasets by linking biological entities that share intended meaning [86]: it is fairly straightforward to create custom URI identifying biological entities such as, for example, *http://flydatabase.org/p53* (flybase:p53) and match it to the equivalent Entrez Gene [87] identifier, *http://www.ncbi.nlm.nih.gov/gene/2768677* (ncbi:276867) by creating the assertion *[flybase:p53 rdfs:seeAlso ncbi:276867]*. Alternatively, if the same identifiers are used, integration is immediate.

Final Introductory Remarks

The application of SW technologies to data-driven biomedical research, even if only partial, can greatly accelerate scientific discovery. It is enough, for example, to obtain an RDF representation of a list of genes derived from a microarray experiment, to have it integrated with a large array of annotations such as gene product or associated diseases [88]. Given the novelty of such technologies, and although there has been extensive community involvement, there is still a long road ahead before SW technologies can fully support the biomedical data acquisition bottleneck in knowledge engineering [12]. There is, for example, a lack of adequate standardized conventions to securely apply SW principles to patient clinical data. Furthermore, the systems that support collaborative ontology edition are detached from the systems that manage and integrate the data elements. As a result, life scientists are often forced to return to conventional organizational schemas, such as spreadsheets and non-interoperable formats, to address their data management needs. A mechanism whereby researchers are free to re-engineer the knowledge representations that best suit their management needs while simultaneously integrating with data available in the Linked Data Cloud [81] is therefore needed. Addressing that challenge comes with the promise of advancing translational research efforts [6] and the immediate application of known disease mechanisms to drug discovery [21].

The work presented here corresponds to the identification of the key requirements for applying semantic web technologies to experimental biomedical data management. With that purpose, the prototypical application S3DB (Simple Sloppy Semantic Database) was developed to fit the requirements of a Linked Data Knowledge Organization System (KOS) for the Life Sciences including the ability to edit the domain representations while instantiating it with data and a flexible fine-grained permission control mechanism. Reflecting our data-driven approach, the technological

requirements and advancements in the S3DB prototype were iteratively devised through direct interaction and validation by biomedical domain experts in multiple areas of expertise including Molecular Epidemiology and Cancer Research.

Chapter 2 | A Knowledge Organization System for the Life Sciences

Chapter Outline

Section 1. J.S. Almeida, C. Chen, R. Gorlitsky, R. Stanislaus, M. Aires-de-Sousa, P. Eleutério, J. Carriço, A. Maretzek, A. Bohn, A. Chang, F. Zhang, R. Mitra, G.B. Mills, X. Wang, and H.F. Deus, "**Data integration gets 'Sloppy'**", *Nature biotechnology*, vol. 24, Sep. 2006, pp. 1070-1. *

It is a well-known fact amongst KE communities that useful models result from an iterative process where the acquisition of new information drives its evolution [4]. Furthermore, early attempts at devising KE systems revealed the need to distinguish between domain knowledge in the form of a terminology (Tbox), and extensional knowledge in the form of assertions (ABox)[89]. Applying the lessons learned from KE principles to the development of a distributed RDF-based data management ecosystem for the Life Sciences we have proposed, by means of a Letter published in the journal Nature Biotechnology, that integration of biomedical data requires biology domain experts to incrementally edit the data model that best describe their data acquisition efforts without compromising functionality. This can be achieved in a scalable manner by relying on the same necessary separation between the Tbox containing the domain descriptors and the Abox containing the data itself as predicted by early KE systems. We open this chapter with the aforementioned Letter.

* The candidate helped identify the Rule/Statement dichotomy, developed the prototype S3DB and validated it with examples from biomedical domains.

Section 2. H.F. Deus, R. Stanislaus, D.F. Veiga, C. Behrens, I.I. Wistuba, J.D. Minna, H.R. Garner, S.G. Swisher, J.A. Roth, A.M. Correa, B. Broom, K. Coombes, A. Chang, L.H. Vogel, and J.S. Almeida, "**A Semantic Web management model for integrative biomedical informatics**", *PloS one*, vol. 3, Jan. 2008, p. e2946. **

The prototypical S3DB tool was then applied in the context of a lung cancer research project involving two institutions in different geographic locations. The proposed challenge was to create a platform that enabled biomedical researchers from both institutions to jointly evolve a distributed data management model that described their acquired data. That exercise revealed a stringent need for addressing data governance and permission management in semantic web KOS, particularly concerning the privacy of patient data. We found support for overlapping data models to be a requirement as well as the ability to finely control the permission that each user should have in individual data elements. Molecular biology findings, for example, which sprouted the collaboration between the groups, were required to remain linked to the confidential patient data from which they were derived. We therefore sought to create an RDFS management model, supported by the prototypical S3DB implementation, to address this set of requirements. The report describing those findings has been included in Section 2.

**The candidate helped identify the components of the S3DB Core Model, implemented the Core Model in the S3DB prototype application, collected the data and representation models

Section 3. J.S. Almeida, H.F. Deus, and W. Maass, "**S3DB core: a framework for RDF generation and management in bioinformatics infrastructures**", *BMC bioinformatics*, vol. 11, Jul. 2010, p. 387. ***

Wider adoption of the S3DB model, by the lung cancer research groups and other biomedical experts, created the need for a clearly defined formalism to support the fine grained permission management requirements that challenged biomedical data management. No accepted standards existed at the time of development to support propagation behaviors that balanced between the two extremes of defining permission at the point of entry or forcing the assignment of permission for every user in every individual data element. A logical framework was therefore devised to support this intended behavior and a set of operators with states were used to describe the relationship between data elements and its usage. This enabled mathematically defined rules governing the propagation of permissions in the S3DB Core Model, not only within a single system, but across multiple systems. The report describing these finding concludes Chapter 2.

***The candidate helped identify the s3db:operators and their intended behaviors based on biomedical use cases, implemented those behaviors in the S3DB prototype and validated the model with examples

we emphatically recommend all users of dbSNP to refer to the 'validation status' tag and use a simple SNP classification scheme, as described above, that aims at extracting RefSNPs with lower error rates. According to our classification, dbSNP (version 124) contains in C1, C2 and C3 2,077,680, 2,946,840 and 3,470,166 entries, respectively. To investigate the differences between those three classes, we extracted the available confidence information. C1 and C2 RefSNPs have higher average values (both 51.4) than SNPs in C3 (43.2, **Supplementary Notes** online). Furthermore, about 87% in C1 and C2 have confidence values of at least 40, in contrast to only 63% in C3 (**Fig. 1d**). As a low confidence value indicates a potential sequencing error, we recommend that bioinformatics and/or experimental efforts either use only C1 and C2 RefSNPs or find a way of excluding from C3 all dbSNP entries with Phred <40 (ref. 11).

Note: Supplementary information is available on the Nature Biotechnology website.

Matthias Platzer[1], Michael Hiller[2],
Karol Szafranski[1], Niels Jahn[1], Jochen Hampe[3], Stefan Schreiber[3], Rolf Backofen[2] & Klaus Huse[1]

[1]Genome Analysis, Leibniz Institute for Age Research–Fritz Lipmann Institute, Beutenbergstr. 11, 07745, Jena, Germany. [2]Institute of Computer Science, Albert-Ludwigs-University Freiburg, Georges-Koehler-Allee 106, 79110 Freiburg, Germany. [3]Institute for Clinical Molecular Biology, Christian-Albrechts-University Kiel, Schittenhelmstr. 12, 24105, Kiel, Germany. e-mail: mplatzer@fli-leibniz.de

1. Kruglyak, L. *Nat. Genet.* **17**, 21–24 (1997).
2. Mitchell, A.A., Zwick, M.E., Chakravarti, A. & Cutler, D.J. *Bioinformatics* **20**, 1022–1032 (2004).
3. Hiller, M. *et al. Nat. Genet.* **36**, 1255–1257 (2004).
4. Valentonyte, R. *et al. Nat. Genet.* **37**, 357–364 (2005).
5. Krawczak, M., Reiss, J. & Cooper, D.N. *Hum. Genet.* **90**, 41–54 (1992).
6. International HapMap Consortium. *Nature* **426**, 789–796 (2003).
7. Korch, C. & Drabkin, H. *Genome Res.* **9**, 588–595 (1999).
8. Stephens, R.M. & Schneider, T.D. *J. Mol. Biol.* **228**, 1124–1136 (1992).
9. Kotlyar, A.B., Borovok, N., Molotsky, T., Fadeev, L. & Gozin, M. *Nucleic Acids Res.* **33**, 525–535 (2005).
10. Ewing, B. & Green, P. *Genome Res.* **8**, 186–194 (1998).
11. Hiller, M. *et al. Am. J. Hum. Genet.* **78**, 291–302 (2006).

Data integration gets 'Sloppy'

To the editor:

Data integration in life sciences currently faces a conundrum[1–4]. On the one hand, the diversity of data is increasing as explosively as its volume. This makes it imperative that some degree of data formatting standardization is agreed upon by the diverse community generating and using that data. On the other hand, the value of individual data sets can only be appreciated when enough of those distinct pieces of the systemic puzzle are put together. Therefore, it is also imperative that standard formats not be enforced so strictly as to be an obstacle to reporting the very novel data that brings value to the targeted systemic integration. We present here a prototype application, termed Simple Sloppy Semantic Database (S3DB), that provides a bridge between loosely structured raw data annotated using personal ontologies and a globally referenceable semantic representation indexed to controlled vocabularies. Wide adoption of this database formalism has the potential to facilitate and optimize data management in a range of research fields, from molecular epidemiology to basic biology.

For most types of biological data, the agreed-upon communal format has a complexity that is far from trivial and requires specialized converters that were not available when the analytical method was first developed. For example, an agreed-upon Minimum Information about Microarray Experiments (MIAME) standard was defined in 2001 (ref. 5), but the jury is still out for much older and widely used techniques such as gel-based proteomics (for example, see ref. 6). Even when, after much consultation, a community standard emerges, the rigidity of minimal descriptions eventually becomes insufficient for stand-alone reposition[7]. Like many others before us, we have reached the conclusion that complementary efforts in proteomics[8], transcriptomics[9] and genomics[10] can only be integrated in a common representation within a semantic framework[2,11]. We have specifically argued[2] for the need to migrate to RDF (Resource Description Framework) from the more widely used XML (Extensible Markup Language) hierarchies or relational structures, a view also espoused by the World Wide Web consortium Life Sciences interest group (http://www.w3.org/2001/sw/hcls/). However, that formalism is cumbersome for configuring information management systems and trades human intuitiveness for machine process expressiveness. This combination of implementation and interface challenges typically loses the very contribution that is needed to put the systemic puzzle together: that of the 'biology domain' expert.

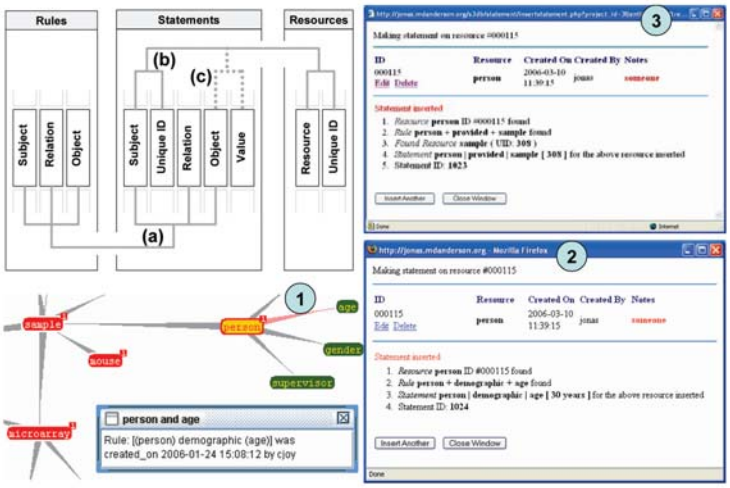


Figure 1 Example of a S3DB application. The indexing scheme is described by the table in the upper left, where the connecting lines identify the three clauses, (a)–(c), verified by the validation engine for a new statement. Three snapshots of the S3DB application for the example discussed in the text are displayed: directed graph depiction of the rules (1), validation log for submission of a literal (nuclear data element such as '30 years') (2) and validation log for the association of two resources (3).

A bridge between poorly structured raw data annotated to personal ontologies and a globally referenceable semantic representation indexed to controlled vocabularies is thus needed. Such a bridge should raise no obstacles to data submission and should instead allow the incremental editing of the underlining data model without compromise of the data already submitted. It should also be deployable as a web-server application such that collaborating users can share a common repository independently of their location. Finally, it should allow the referencing to external controlled vocabularies and should be exportable as RDF[2]. With this in mind, we have developed a prototype application, S3DB, that incorporates all these characteristics.

The proposed implementation of editable semantic reposition relies on a relational backbone made of three tables: rules, statements and resources. The concept driving the configuration of S3DB is purely semantic and relies solely on documenting an entity relationship (ER) model[12] of the type *[Subject][Property][Object]*. The solution that enables editable data model reposition consists of an indexing scheme where the permanence of the indexes rather than the permanence of the element names allows renaming and reassociation without loss of content. Although the relational backbone of S3DB consists of three tables, they do not establish a relational model. Instead, their interoperation relies on a validation engine that checks for syntactic and semantic consistency. Data are submitted as statements made of five element vectors, *[Subject] [UID] [Property] [Object] [Value]* that are verified for the following: (i) that the triple *[Subject][Property][Object]* exists in the rules table; (ii) that the resource unique index (UID) pair *[Subject][UID]* exists in the resources table; and (iii) that if *[Object]* is a resource (if it is declared as having UID in rules) then *[Value]* has to be a valid UID for that resource (i.e. , the pair *[Subject][UID]* exists in the resources table).

This solution mimics locally the sort of evolution of the data model that we expect to achieve for global representations using RDF[2]. Its workings are illustrated using an example in **Figure 1**. To assign an age to a patient, the first step is to add that property to the domain of discourse (for example, to include an entry in the rules table saying that people have age as a demographic property; see popup window inset). Subsequently, a statement can be made for an existing patient, UID 115, saying that she is 30 years of age (literal value) and then that a sample, UID 308, was collected. Because not just all resources, but also all rules and statements, are uniquely indexed (UID), their contents can subsequently be edited for renaming of resources and rewiring of relations. The result is a conveyor belt of successive editing into more structured and global formats. The S3DB prototype was developed with open-source languages and is made freely available with open source for unrestricted use and modification (http://www.s3db.org).

ACKNOWLEDGEMENTS

This work was partially funded by the National Heart, Lung and Blood Institute of the US National Institutes of Health, under contract no. N01-HV-28181, and the PREVIS project (Pneumococcal Resistance Epidemicity and Virulence, An International Study), contract number LSHM-CT-2003-503413 from the European Commission.

Jonas S Almeida[1,3], Chuming Chen[2], Robert Gorlitsky[2], Romesh Stanislaus[2], Marta Aires-de-Sousa[3], Pedro Eleutério[3], João Carriço[3], António Maretzek[3], Andreas Bohn[3], Allen Chang[1], Fan Zhang[4], Rahul Mitra[4,5], Gordon B Mills[4], Xiaoshu Wang[2] & Helena F Deus[3]*

[1]Department of Biostatistics and Applied Mathematics, University of Texas, 1515 Holcombe Blvd., Box 0447, Houston, Texas 77030-4009, USA. [2]Department of Biostatistics, Bioinformatics & Epidemiology, Medical University of South Carolina, 135 Cannon Street, Suite 303, Charleston, South Carolina 29425, USA. [3]Instituto de Tecnologia Química e Biológica da Universidade Nova de Lisboa (ITQB/UNL), Av. da República (EAN), 2781-901 Oeiras, Portugal. [4]Kleberg Center for Molecular Markers, Department of Molecular Therapeutics, University of Texas M. D. Anderson Cancer Center, 1515 Holcombe Blvd., Box 0317, Houston, Texas 77030-4009, USA.
e-mail: jalmeida@mdanderson.org

1. Hey, T. & Trefethen, A.E. *Science* **308**, 817–821 (2005).
2. Wang, X., Gorlitsky, R. & Almeida, J.S. *Nat. Biotechnol.* **23**, 1099–1103 (2005).
3. Buetow, K.H. *Science* **308**, 821–824 (2005).
4. Foster, I. *Science* **308**, 814–817 (2005).
5. Brazma, A. *et al. Nat. Genet.* **29**, 365–371 (2001).
6. Stanislaus, R. *et al. BMC Bioinformatics* **5**, 9 (2004).
7. Shields, R. *Trends Genet.* **22**, 65–66 (2006).
8. Stanislaus, R. *et al. Bioinformatics* **21**, 1754–1757 (2005).
9. Almeida, J.S. *et al. Compar. Func. Genomics* **6**, 132–137 (2005).
10. McKillen, D.J. *et al. BMC Genomics* **6**, 34 (2005).
11. Neumann, E. *Sci. STKE* **2005**, pe22 (2005).
12. Chen, P.P.S. *Assoc. Comput. Machinery Trans. Database Syst.* **1**, 9–36 (1976).

Replacing cRNA targets with cDNA reduces microarray cross-hybridization

To the editor:

Gene-expression microarrays are designed to measure relative concentrations of transcripts through the specific hybridization of an immobilized DNA probe to its complementary target. This technology is viable to the extent that a single, rather permissive hybridization condition allows most probes to bind specifically to their targets. Despite efforts to maximize stringency, a significant hybridization signal can still be detected on various oligonucleotide-based platforms, even when there are a few mismatches between probe and target[1–3]. Furthermore, several groups have detected widespread cross-hybridization in microarray measurements[4,5], and on the order of 10% of the probes on a common oligonucleotide array platform were predicted to be susceptible to cross-hybridization[5]. Efforts to optimize probe length found that longer probes enjoy stronger signal intensity but also suffer from increased propensity toward cross-hybridization[6]. Therefore, nonspecific binding remains a significant source of measurement error and may be the reason why quantitative reverse transcription (qRT)-PCR fails to confirm about 10–20% of difference calls made by microarray analysis (reviewed in ref. 7). Here, we report that a high-level of promiscuity in DNA-RNA hybridization underlies widespread cross-hybridization in microarrays. This cross-hybridization can be reduced using cDNA targets in place of cRNA.

From its inception, microarray technology took advantage of either of two types of biochemical entities as the labeled target, cRNA[8] or cDNA[9]. Although in many aspects these two types of labeled target are considered to be equivalent for the purpose of microarray analysis, the use of cRNA has held an important methodological advantage. Because RNA polymerase does not require a primer, it was rather straightforward to design a near-linear

OPEN ACCESS Freely available online

PLoS one

A Semantic Web Management Model for Integrative Biomedical Informatics

Helena F. Deus[1,2], Romesh Stanislaus[1], Diogo F. Veiga[1], Carmen Behrens[3], Ignacio I. Wistuba[3,4], John D. Minna[5], Harold R. Garner[5,6,7,8,9], Stephen G. Swisher[10], Jack A. Roth[10], Arlene M. Correa[10], Bradley Broom[1], Kevin Coombes[1], Allen Chang[1], Lynn H. Vogel[1,11], Jonas S. Almeida[1]*

1 Department of Bioinformatics and Computational Biology, The University of Texas M.D. Anderson Cancer Center, Houston, Texas, United States of America, **2** Instituto de Tecnologia Química e Biológica, Universidade Nova de Lisboa, Lisboa, Portugal, **3** Department of Thoracic/Head and Neck Medical Oncology, The University of Texas M.D. Anderson Cancer Center, Houston, Texas, United States of America, **4** Department of Pathology, The University of Texas M.D. Anderson Cancer Center, Houston, Texas, United States of America, **5** Hamon Center for Therapeutic Oncology Research, Simmons Cancer Center, University of Texas Southwestern Medical Center, Dallas, Texas, United States of America, **6** Department of Internal Medicine, University of Texas Southwestern Medical Center, Dallas, Texas, United States of America, **7** Eugene McDermott Center for Human Growth and Development, University of Texas Southwestern Medical Center, Dallas, Texas, United States of America, **8** Center for Biomedical Inventions, University of Texas Southwestern Medical Center, Dallas, Texas, United States of America, **9** Department of Biochemistry, University of Texas Southwestern Medical Center, Dallas, Texas, United States of America, **10** Department of Thoracic and Cardiovascular Surgery, The University of Texas M.D. Anderson Cancer Center, Houston, Texas, United States of America, **11** Department of Biomedical Informatics, Columbia University, New York, New York, United States of America

Abstract

Background: Data, data everywhere. The diversity and magnitude of the data generated in the Life Sciences defies automated articulation among complementary efforts. The additional need in this field for managing property and access permissions compounds the difficulty very significantly. This is particularly the case when the integration involves multiple domains and disciplines, even more so when it includes clinical and high throughput molecular data.

Methodology/Principal Findings: The emergence of Semantic Web technologies brings the promise of meaningful interoperation between data and analysis resources. In this report we identify a core model for biomedical Knowledge Engineering applications and demonstrate how this new technology can be used to weave a management model where multiple intertwined data structures can be hosted and managed by multiple authorities in a distributed management infrastructure. Specifically, the demonstration is performed by linking data sources associated with the Lung Cancer SPORE awarded to The University of Texas MDAnderson Cancer Center at Houston and the Southwestern Medical Center at Dallas. A software prototype, available with open source at www.s3db.org, was developed and its proposed design has been made publicly available as an open source instrument for shared, distributed data management.

Conclusions/Significance: The Semantic Web technologies have the potential to addresses the need for distributed and evolvable representations that are critical for systems Biology and translational biomedical research. As this technology is incorporated into application development we can expect that both general purpose productivity software and domain specific software installed on our personal computers will become increasingly integrated with the relevant remote resources. In this scenario, the acquisition of a new dataset should automatically trigger the delegation of its analysis.

Citation: Deus HF, Stanislaus R, Veiga DF, Behrens C, Wistuba II, et al. (2008) A Semantic Web Management Model for Integrative Biomedical Informatics. PLoS ONE 3(8): e2946. doi:10.1371/journal.pone.0002946

Editor: Eshel Ben-Jacob, Tel Aviv University, Israel

Received March 25, 2008; **Accepted** July 12, 2008; **Published** August 13, 2008

Funding: This work was supported in part by the National Heart, Lung and Blood Institute (NHLBI), by the National Cancer Institute (NCI) of the US National Institutes of Health (NIH), and by the Center for Clinical and Translational Sciences under contracts no. N01-HV-28181, P50 CA70907, and 1UL1RR024148, respectively. The authors also acknowledge support by the PREVIS project, contract number LSHM-CT-2003-503413 from the European Union Commission.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: jalmeida@mdanderson.org

Introduction

Data management and analysis for the life sciences

"The laws of Nature are written in the language of mathematics" famously said Galileo. However, in recent years efforts to analyze the increasing amount and diversity of data in the Life Sciences has been correspondingly constrained not so much by our ability to read it as by the challenge of organizing it. The urgency of this task and the reward of even partial success in its accomplishment have caused the interoperability between diverse digital representations to take center stage [1–5]. Presently, for those in the Life Sciences enticed by Galileo's pronouncement, the effort of collecting data is no longer focused solely on field/bench work. Instead, it often consists of painfully squeezing the pieces of the systemic puzzle from the digital media where the raw data is held hostage[6]. It is only then that a comprehensive representation amenable to mathematical modeling really becomes available[7]. This is not a preoccupation exclusive to the Life Sciences. Integration of software applications is also the driving force behind new information management systems architectures that seek to eliminate the boundaries to interoperability between data and services. This preoccupation indeed

underlies the emergence of service oriented architectures [8–11], even more so in its event driven dynamic generalization [12]. It also underlies the development of novel approaches to software deployment (Figure 1) that juggle data structures between server and client applications. Presently, a particularly popular design pattern is the usage-centric Web 2.0 [13,14] which seeks a delicate balance in the distribution of tasks between client and server in order to diminish the perception of a distinction between local and remote computation.

Semantic web technologies [3,15–21] represent the latest installment of web technology development. In what is being unimaginatively designated as Web 3.0[22,23], a software development design pattern is proposed where the interoperability boundaries between data structures, not just between the systems that produce them, is set to disappear. The defining characteristic of this environment is that one can retrieve data and information by specifying their desired properties instead of explicitly (syntactically) specifying their physical location. The desirability of this design can clearly be seen in systems in which clinical records are matched with high throughput molecular profiles, each of which stem from very distinct environments and are often the object of very different access management regulations.

Inadequacy of conventional systems for Translational Research

On the one hand, high throughput molecular Biology core facilities and improved medical record systems are able to document individual data elements with increasing detail. On the other hand, researchers producing the data and models that critically advance the understanding of biological phenomena are increasingly separated from their use by the specialization inherent in each of these activities. Consequently, bridging between the information systems of basic research and their clinical application becomes a necessary foundation for any translational exploits of new biomedical knowledge[3,24]. The alternative, using conventional data representations where the data models cannot evolve, typically requires the biomedical community to complement the data representation with a clandestine and inefficient flurry of datasets exchanged as spreadsheets through email.

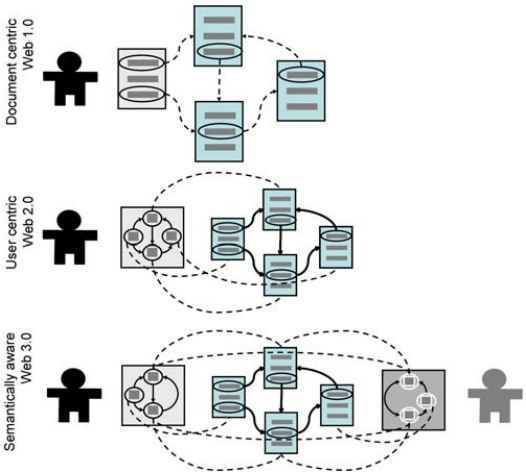


Figure 1. Three generations of design patterns for web-based applications. The original design ("1.0") consists of collections of hypertext documents that are syntactically (dashed lines) interoperable (traversing between them by clicking on the links), regardless of the domain content. The user centric web 2.0 applications use internal representations of the external data structures. This representation is asynchronously updated from the reference resources which are now free to have a specialized interoperation between domain contents. An example of this approach is that followed by AJAX-based interfaces. Finally, the ongoing emergence of the semantic web promises to produce service oriented systems that are semantically interoperable such that the interface application reacts to domains of knowledge specifically. At this level all applications tend to be web-interoperable with peer-to-peer architectures complementing the client-server design of w1.0 and w2.0.
doi:10.1371/journal.pone.0002946.g001

Foundations for a novel solution

As others before us[5], we have argued previously for the use of semantic web formats as the foundation for developing more flexible and articulated data management and analytical bioinformatics infrastructures[20]. A software prototype was then produced following those technical specifications to provide a flexible web-based data sharing environment within which a management model can be identified[24]. In this third report we describe the resulting core model supporting distributed and portable data representation and management. In practice this translates into a small application deployed in multiple locations rather than a large infrastructure at a single central location. The open source prototype application described here has been made public[25]. All deployments support a common data management and analysis infrastructure with no constraints on the actual data structures described.

A very brief history of data

The formatting of data sets as portable text mirrors the same three stages described for web-based applications in Figure 1. As described in Figure 2, data representation has been evolving from tabular text formats ("flat files"), to self described hierarchical trees of tags (extended markup languages, XML), and finally to the subject-predicate-object triples of Resource Description Framework (RDF)[26]. We have been active participants in these transformations [24,27,28], and like many others concluded that in order to bridge the fragmentation between distinct data structures, we needed to break down the data structures themselves[20], that is, to reduce the interoperable elements to RDF triples[29]. In addition to its directed labeled graph nature, RDF formats[29] have a second defining characteristic: each of the three elements has a Uniform Resource Identifier (URI), which, for the purposes of this very brief introduction, can be thought as a unique locator capable of directing an application to the desired content or service. It is also interesting to note that at each level of this three-stage progression (Figure 2) we find data elements that have "matured", that is, that present a stable representation which remains useful to specialized tools. When this happens we find that those elements remain convenient representations preserved whole within more fragmented formats. For example, we find no advantages in breaking down mzXML[30] representations of mass spectrometry based proteomics data. Instead, these data structures are used as objects of regular RDF triples. The mzXML proteomics data structure offers an paradigmatic illustration of the evolution of ontologies as efforts to standardize data formats[31]. It would be interesting to understand if the lengthy effort headed by the Human Proteomics Organization, HUPO, to integrate it reflects the difficulty to justify reforming[32] a representation that remains useful[33].

The advancement towards a more abstract, more global and more flexible representation of data is by no means unique to the Life Sciences. However, because of the exceptional diversity of

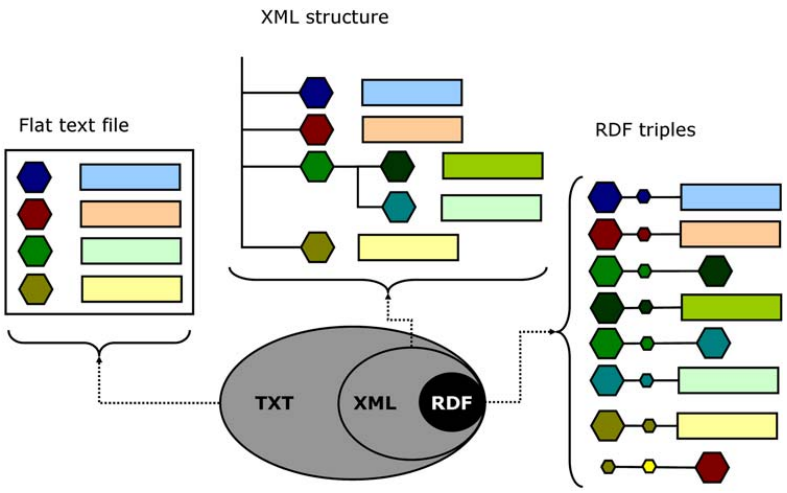


Figure 2. Evolution of formats for individual datasets. Hexagons, rectangles and small circles indicate data elements, respectively, attributes, their values, and relations. First, flat file formats such as fasta or the GeneBank data model were proposed to collect attribute-value pairs about an individual data entry. The use of tagging by extended markup languages (XML) allowed for the embedding of additional detail and further definition of the nature of the hierarchical structure between data elements. More recently, the resource description framework (RDF) further generalized the XML tree structure into that of a network where the relationship between resources (nodes) is a resource itself. Furthermore, the referencing of each resource by a unique identifier (URI) implies that the data elements can be distributed between distinct documents or even locations.
doi:10.1371/journal.pone.0002946.g002

that domain's fluidity, the Life Sciences are where the Semantic Web may find its most interesting challenge and as well, hopefully, where it will find its most compelling validation[15].

Mathematics for data models

It has not been lost to the swelling ranks of Systems Biologists that the reduction of data interoperability to the ternary representation of *relations* [34] brings the topic solidly back to the Galilean fold of Mathematics as a language. The reduction of data structures to globally referenced dyadic relations (functions of two variables), such as those of the Entity-Relationship (ER) model, brings in rich feeds from the vein of Logic. In the process, and beyond Galileo's horizon, assigning a description logic value[35–37] to some RDF predicates (for example, specifying that something is part of or, on the contrary, is distinct from something else) allows the definition of procedures. This further elaboration of RDF has the potential to transform data management into an application of knowledge engineering, and more specifically of artificial intelligence (AI). This reclassification reflects the dilution of the distinction between data management and data analysis that is apparent even in an introduction as brief as this one. Another clear indication of this transformation is that it re-ignites the opposition between data-driven and rule-driven designs for semantic web representation[38–42], a recurring topic in AI. It is important to note that the management model proposed here is orthogonal to that discussion. Its purpose is solely to enable the distribution[43] of a semantic data management system that can withstand changes in the domain of discourse, independently of the rationale for the changes themselves.

Software engineering for Bioinformatics

This overview of modern trends in integrative data management is as significant for what is covered as for what is missed – what management models should be used to control the generation and transformation of the data model? It is interesting to note that the management models that associate access permissions with the population of a data model have traditionally been the province of software engineering. This may at first appear to be a reasonable solution. Since instances of a data structure in conventional databases are contained in a defined digital media, permission management is an issue of access to the system itself. However, this ceases to be the case with the semantic web RDF triples because they weave data structures that can expand indefinitely between multiple machines. Presently, the formalisms to manage data in the semantic web realm are still in the early stages of development, notably by the World Wide Web consortium (W3C) SKOS initiative (Simple Knowledge Organization Systems). This initiative recently issued a call[44] for user cases where good design criteria can be abstracted and recommendations be issued on standard formats. As expected[15], the Life Sciences present some of the most convoluted user cases in which a multitude of naïve domain experts effectively need to maintain data structures that are as diverse and fluid as the experimental evidence they describe[24].

Materials and Methods

The most extreme combination of heterogeneous data structures and the need for very tight control of access is arguably found in applications to Personalized Medicine, such as those emerging for cancer treatment and prevention. At the Univ. Texas MDAnderson Cancer Center at Houston and the Southwestern Medical Center at Dallas we have deployed the S3DB semantic web prototype to engage the community of translational researchers of the University of Texas Lung Cancer SPORE [45] in identifying a suitable management model. This exercise involved over one hundred researchers and close to half a million data entries, of clinical and molecular nature. Right at its onset integrating access permissions in the definition of the data models was identified as an absolute necessity by the participants, as

anticipated by the SKOS group. As a consequence, a data driven "core model", S3DBcore, that accommodates management specifications as part of data representation, was developed and is described here. The software used is provided with open source at www.s3db.org. Only open source tools were used in development of this web-based web-service: PHP 5 was used for server side programming and both MySQL and PostgreSQL were tested as the relational backbone for PHP's database abstraction class. At the same location detailed documentation about S3DB's Application Programming Interface (API) is also provided.

Results

Units of representation

The most fundamental representation of data is that of attribute-value (AV) pairs, for example, <color,"blue">. The generic data management infrastructure proposed here can be described as that of encapsulating AV pairs through the use of another fundamental unit of representation, the Entity-Relation-Entity model (ER), such as <sky, has, color>. Each entity can then be associated with one or more AV pairs using the entity-attribute-value EAV model, for example, <sky, color, "blue">. Fast forwarding three decades of computer science and knowledge engineering and we reach the present day development of a representation framework where each element of the triple is a resource with a unique identifier, with the third element of the triple having the option of being a literal, that is, of having an actual value rather than a placeholder. This single sentence very broadly describes the Resource Description Framework (RDF) which is at the foundation of the ongoing development of the Semantic Web[29], just like hypertext (HTML) was the enabling format for the original Web. It is important to note that the evolution of representation formats typically takes place through generalization of the existing ones. For example, extended markup language-based files (XML) are still text files, and RDF documents are still XML structures (Figure 2). As noted earlier, this succession is closely paralleled by refinements of software design patterns (Figure 1). This reification process is often driven by the necessity to maintain increasingly complex data at a simpler level of representation where they remain intelligible for those who generate and use the data. Accordingly, in the next section triple relations will be weaved around the AV pair with that exact purpose: to produce a core model that is simple enough to be usable by naïve users that need to interact with heterogeneous data hosted in a variety of machines (Figure 3), yet sophisticated enough to support automated implementation.

Weaving a distributed information management system

The objective of this exercise is to produce a data management model that can be distributed through multiple deployments of the Database Management Systems (DBMS) which implies a mechanism for migration access permissions. Simultaneously, this model should allow different domain experts to evolve their own data models without compromising pre-existing data. Achieving these

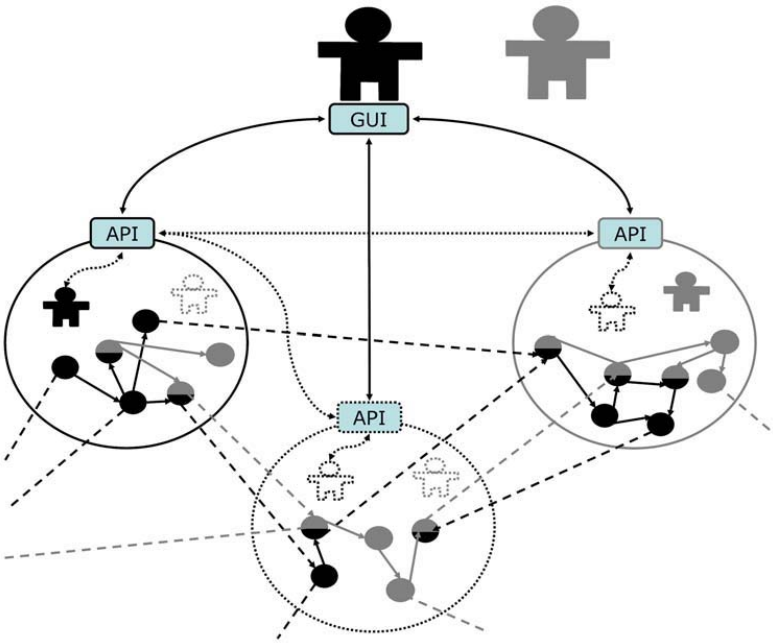


Figure 3. Illustration of the desirable functionality: distinct users, with identities (solid icon) managed in distinct S3DB deployments (circular compartments), which they control separately, share a distributed and overlapping data structure (arrows between symbols) that they also manage independently: some data elements are shared (mixed color symbols) others are not. This will require the identity verification to propagate between deployments peer-to-peer (P2P, dotted lines), including to deployments where neither user maintains an identity (dotted circular compartment). This is in contrast with the conventional approach of having distinct users manage insular deployments with permissions managed at the access point level.
doi:10.1371/journal.pone.0002946.g003

two goals simultaneously can only be realized if the proposed distributed system is composed of node applications that are not only syntactically interoperable, but also semantically transparent. For a discussion of the absolute need for evolvable data models in the Life Sciences see [24]. That report is also where the DBMS prototype, S3DB, was first introduced (version 1.0). Finally, the Application Programming Interface (API) needs to support the semantic interoperability in a way that spans multiple deployments (Figure 3). The data model developed to achieve these goals is described in Figure 4.

A Core data management model that is universal and distributed

The directed labeled graph nature of RDF triples, coupled with their reliance on unique identifiers (as URIs), enables data structures to be scattered between multiple machines while permitting different domains of discourse to use the same data elements differently. However, those two characteristics alone do not address the management issue: how to decide when, where and what can be viewed, inserted, deleted and by whom. It is clear that the conventional approach of dealing with permissions at the level of access to the data store is not appropriate to the Life Sciences[5] where multiple disciplines and facilities are contributing to a partially overlapping representation of the system. It cannot be overstated that this is particularly the case when the system is designed to host clinical data. To solve this problem we have developed a core data model where membership and permission can migrate with the data. We have also developed a prototype application to support such a distributed data management system (Figure 3), which we make freely available with open source[25].

Discussion

The proposed core model is detailed in Figure 4 and will be now discussed in more detail. This diagram is best understood chronologically, starting with the very basic and nuclear collection

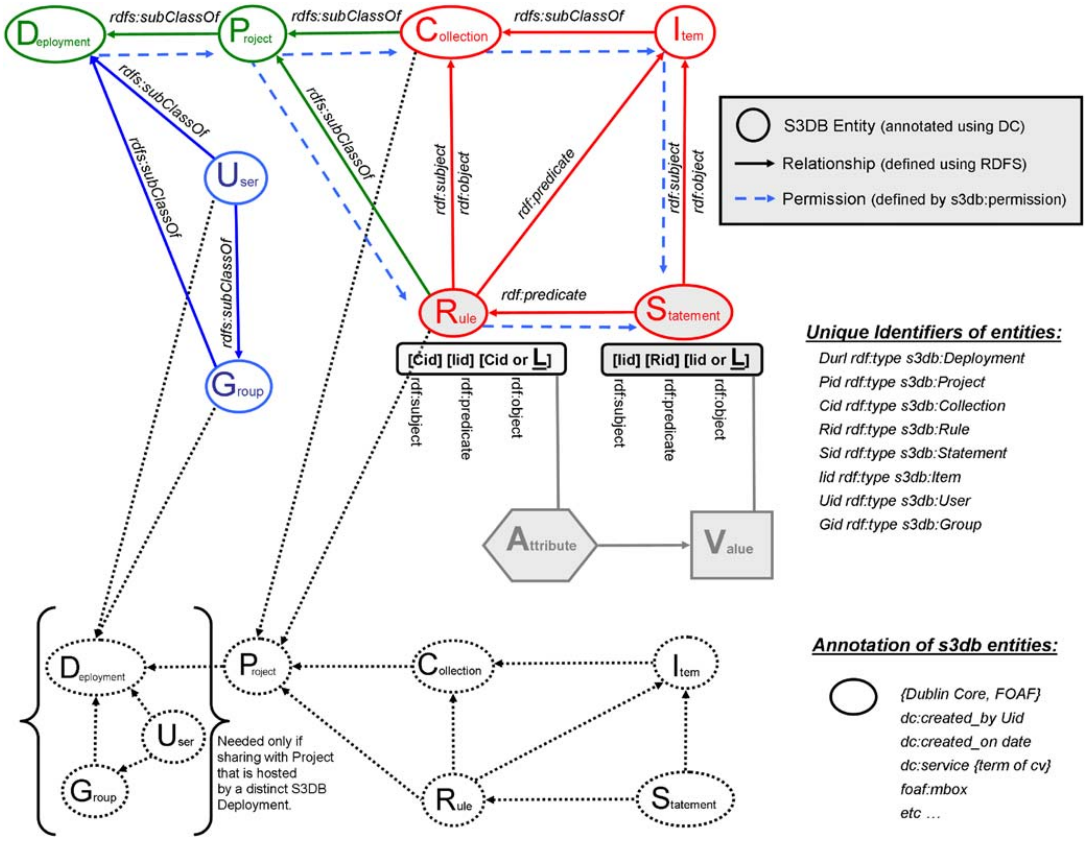


Figure 4. Core model developed for S3DB (supported by version 3.0 onwards). This diagram can be read starting from the most fundamental data unit, the Attribute-Value pair (filled hexagonal and square symbols). Each element of the pair is object of two distinct triples, one describing the domain of discourse, the *Rules*, and the other made of *Statements* where that domain is populated to instantiate relationships between entities. The latter includes the actual Values. Surrounding these two nuclear collection of triples, is the resolution of *Collection* and its instantiation as *Item* that define the relationship between the individual elements of *Rules* and *Statements*. The resulting structure is then organized in *Projects* in such a way that the domain of discourse can nevertheless be shared with other *Projects*, in the same or in a distinct deployment of S3DB. Finally, a propagation of user permissions (dashed line) is defined such that the distribution of the data structures can be traced. See text for a more detailed description.
doi:10.1371/journal.pone.0002946.g004

of attribute-value pairs and then proceeding to their encapsulation by three consecutive layers – the semantic schema, assignment of membership and, finally the permission propagation.

Schema

The first layer of encapsulation is the definition and use of a domain of discourse (elements in red in Figure 4). This was achieved in typical RDF fashion by defining two sets of triples, one defining a set of rules and the second, the statements, using them. As discussed elsewhere [24], there are good reasons to equip those who generate the data with the tools to define and manage their own domains of knowledge. The ensuing incubation of experimental ontologies was facilitated by an indexing scheme that mimics the use of subject, verb, object in natural languages. This indexing is achieved by recognizing *Collections* and the *Items* they contain as elements of the two sets of nuclear triples (*Rules* and *Statements*).

Organization

The second layer of formal encapsulation corresponds to the assignment of membership. This process extends the designation of *Items* in the previous level, by assigning the *Collections* that contain them and *Rules* that relate them to *Projects* that are hosted by individual *Deployments* of the prototype S3DB application. In the diagram, the membership dependencies are accordingly labeled as *rdfs:subClassOf* [29]. Note that memberships can also be established with remote resources (dotted lines in Figure 4), that is, between resources of distinct deployments. Defining remote memberships presents little dificulty in the RDF format because each element of the triple is refered to by a universal identifier (a URI), unique accross deployments. On the other hand, managing permission to access the remote content is a much harder problem, which we will address by supporting migration of identity. The alternative solution to migration of identities is migrating the contents along membership lines. However, that was, unsurprisingly, found to be objectionable by users with a special attention to privacy and confidentiality issues. It would also present some logistic challenges for larger datasets. In contrast, the definition of a temporary, portable, identity key or token needed for migration of identity is typically incommensurably smaller than the content it permits access.

Permissions

The final layer of encapsulation defines *Users* and *Groups* within *Deployments* and controls their permissions to the data (blue in Figure 4). As with rest of the core model, the identification of proposed management of permissions was directed by user cases. That exercise determined that user identities should be maintained by specific Deployments of S3DB but also that they may be temporarily propagated to other deployments. That solution, illustrated in Figure 3, allows one application to request the verification of an identity in a remote deployment, which then verifies it in the identity's source deployment and assigns it a temporary key or token, say, for one hour. All that is propagated is a unique alphanumeric string, the temporary token, paired with the user's URI. No other user information is exchanged. As a consequence, for the remainder of the hour, the identification will be asynchronously available in both deployments, which enables the solution described in Figure 3, where a single interface can manipulate multiple components of a large, distributed systems level representation of the target data. Interestingly, because the multiple deployments of S3DB are accessed independently by multiple deployments of various applications, the mode of syntactic interoperation is *de facto* peer-to-peer. The propagation of permissions flows in the sequence indicated by the dashed blue lines in Figure 4. When a permission level is not defined for a resource, say for a *Item*, then it is borrowed from the parent entity, in this example, from the corresponding *Collection*. When there is a conflict then the most restrictive option is selected. For example a conflict can arise for a *Statement* which inherits permissions from both *Rules* and *Collections*. Another frequent example happens when a user belongs to multiple groups with distinct permissions to a common target resource.

Permission management is a particularly thorny issue in life sciences applications because of the management of multiple data provenances. Relying on distributed hosting of the complementary data sources compounds the management of multiple permissions even further because it also involves multiple permission management systems. Finally, permission management is often treated *ad hoc* by the management systems themselves where it is resolved as access permission to the system as a whole rather than being specified in the data representation. Because each source often describes a specialized domain, it is guarded with understandable zeal. We argue here that propagation of permissions is the only practical solution to determine how much information is to be revealed in different contexts. Consequently, whereas the relationships between the 8 S3DB entities (oval symbols in Figure 4) are defined using RDF schema[26] (RDFS), and their tagging uses the well established Dublin Core[46], the permission propagation layer is a novel component of the proposed management model. In order to respond to widest range of the user cases driving model identification, the propagation was defined by three parameters, view, edit, and use. Each of these parameters can have three values, 0, 1 or 2, corresponding to, respectively, no permission, permission only on entries submitted by the user, and permission on all entries of that resource. *Users* and *Groups* (blue entities in Figure 4) can have these three types of permissions on *Projects*, *Collections*, *Rules*, *Items* and *Statements*. Among those five entities, additional permissions can be issued, for example, a *Project* may have specific permissions on *Collections* and *Rules*. *Collections* may have further permissions on their *Items*. The same reasoning, in reverse, establishes what should happen when permission is not specifically defined for a given entity. For example, for a *Statement* the permission would be inherited from the parent entities, *Item* and *Rule*. If those two entities did not specify specific permissions for the target statement, then those are searched upstream (Figure 4) until reaching the *Project* or even *Deployment* level. According to this mechanism, the conventional role of a system administrator corresponds to a user with permissions 222 at Deployment level. It is worth recalling that propagation of permissions between data elements in distinct S3DB deployments happens through the sharing the membership in external *Collections* and *Rules* (dotted lines), not through extending the permission inheritance beyond the local deployment. This is not a behavior explicitly imposed on the distributed deployment; it emerges naturally from the fact that *Rule* sharing specifies a permission which, remote or local, interrupts the permission inheritance. In practice both the user of the interface and the programmer using the API can ignore the intricacies of this process, which was identified to be the intuitive, sensible, propagation of permissions that we found naïve users to expect in user-case exercises.

Portability

This discussion would not be complete without unveiling some defining technical details about how portability is addressed by this design. So far we have been loosely equating "unique identifiers" with the use of Uniform Resource Identifiers (URI). More specifically, the right hand side of Figure 4 includes a list of eight

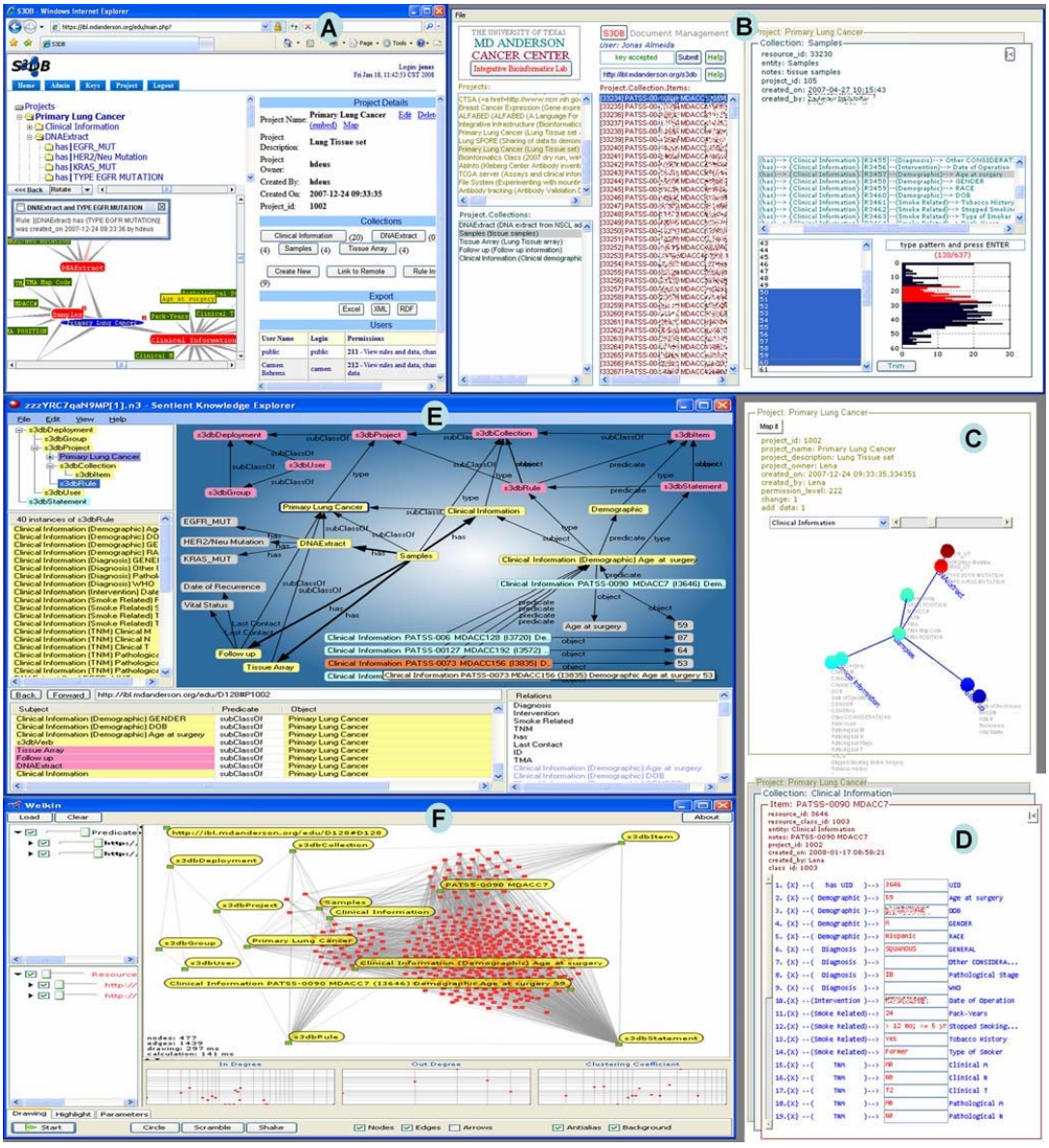


Figure 5. Snapshots of interfaces using S3DB's API (Application Programming Interface). These applications exemplify why the semantic web designs can be particularly effective at enabling generic tools to assist users in exploring data documenting very specific and very complex relationships. Snapshot A was taken from S3DB's web interface, which is included in the downloadable package[25]. This interface was developed to assist in managing the database model and, therefore, is centered on the visualization and manipulation of the domain of discourse, its *Collections of Items* and *Rules* defining the documentation of their relations. The application depicted on snapshots B–D describe a document management tool S3DBdoc, freely available as a Bioinformatics Station module (see Figure 6). The navigation is performed starting from the Project (C), then to the *Collection* (B) and finally to the editing of the *Statements* about an *Item* (D). The snapshot B illustrates an intermediate step in the navigation where the list of *Items* (in this case samples assayed by tissue arrays, for which there is clinical information about the donor) is being trimmed according to the properties of a distant entity, Age at Diagnosis, which is a property of the Clinical Information *Collection* associated with the sample that originated the array results. This interaction would have been difficult and computationally intensive to manage using a relational architecture. The RDF formatted query result produced by the API was also visualized using a commercial tool, Sentient Knowledge Explorer (IO-Informatics Inc), shown in snapshot E, and by Welkin, developed by the digital inter-operability SIMILE project at the Massachusetts Institute of Technology. See text for discussion of graphic representations by these tools. To protect patient confidentiality some values in snapshots B and D are scrambled and numeric sample and patient identifiers elsewhere are altered.

doi:10.1371/journal.pone.0002946.g005

types of locally unique identifiers that can be assigned to the same number of entities that define the core model. It is easy to see how this indexing can be made globally unique by concatenating them with the *Deployment*'s ID, itself unique, for example using its URL. Indeed this is what is supported by the accompanying prototype software, with a generalizing twist with very significant consequences: *Did* can either be the deployment address or anything that indicates what that address is. For example, it can indicate an HTML document or even an entry in a database where this address is specified. More interestingly, it can also be a simple alphanumeric code that is maintained at www.s3db.org in association with the actual URL of the target deployment. The flexible global indexing achieved by either scenario allows the manipulation of entire databases management systems as portable data structures. It also allows for novel management solutions through manipulation of the DBMS logical structure. For example, defining a *Did* as 'localhost' would have the effect of severing all logical connections to any usage outside that of the server machine. None of these more fanciful configurations were validated with the Lung Cancer SPORE user community even if they are fully supported by the accompanying prototype. Nevertheless, its possibility enables some interesting scenarios for data management and indeed for Knowledge Engineering.

User Interfaces

The ultimate test for a data management model is the intuitiveness of what it communicates through the user interface[47,48]. The structure of S3DBcore offers some useful guidelines in this regard. The experimental values are represented in a combination of *Items* and *Statements* (Figure 4). There are two routes to that endpoint. One possibility is to take the document management approach of navigating from *Projects* to *Collections*, then to their *Items* and finally to the *Statements*. This is the scenario that will suit data centric activities such as querying and updating existing data or inserting new data. A real, working example of how that interface may look is depicted in Figure 5-B, which details an intermediate step between selecting a *Project* (Figure 5-B), and identifying and manipulating an individual entry made of multiple statements about an *Item* (Fig. 5-D). The mechanism used to distribute rich graphics applications and their interoperation with S3DB is detailed in Figure 6. Another possibility is to navigate from the *Project* to the collection of *Rules*, most likely represented as a directed labeled graph network, and then browse the *Statements* as an instantiation of the *Rules*, exemplified by another snapshot of a working application, Figure 5-A. This application is the standard web-based user interface distributed with S3DB package[25]. Unlike the bookkeeping approach of the document centric model (Figure 5-B), the rule centric view (Figure 5-A) is most suitable to investigate the relationship between different parts of the domain of knowledge and to incubate[24] a more comprehensive and exact version of the ontology. However, and this may be the most relevant point, since S3DB's API returns query results as RDF, any RDF browser can be used to explore it. This point is illustrated in figures 5E and F where, respectively, a commercial semantic web knowledge explorer (Sentient, IO-Informatics Inc) and Welkin, a popular RDF browser developed at the Massachusetts Institute of Technology, are use to visualize the same S3DB Lung Cancer project depicted in Figs. 5A and B. Whereas the former is designed as a tool for knowledge discovery, the latter offers a global view of distributed data structures. The value of the core model described in Figure 4 as a management template for individual data elements will be apparent upon close inspection of Fig. 5E. The different colors, automatically set by Sentient KE, distinguish the core model (pink), where permission management takes place, from the instantiation of

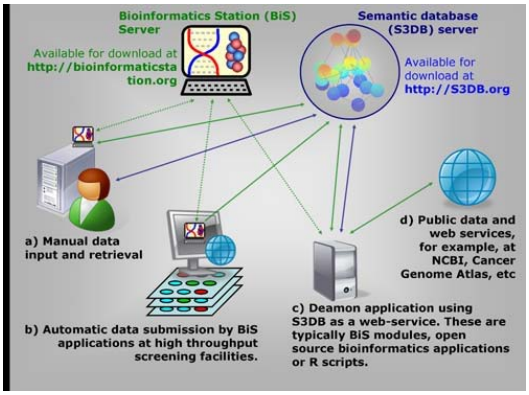


Figure 6. Prototype infrastructure for integrated data management and analysis being tested by the Univ. Texas Lung cancer SPORE. The system is based on two components, a network of universal semantic database servers and a code distribution server that delivers applications in response to the use of ontology. Four distinct user cases are represented, a–d, which rely on a combination of download of interpreted code (green arrows) or direct access to web-based graphic user interfaces or web-based API (blue arrows, in the latter case using Representational State Transfer, REST). The dotted lines represent regular updating of the application, propagating improvements in the application code.
doi:10.1371/journal.pone.0002946.g006

their entities, in yellow. These two layers describe the context for individual entries specifying the age at surgery of 5 patients. The same display includes access to molecular work on tumor samples, in this case using tissue arrays and DNA extracts. The distinct domains are therefore integrated in an interoperable framework in spite of the fact that they are maintained, and regularly edited, by different communities of researchers. As a consequence, the database can evolve with the diversification of data gathering methodologies and with the advancement in understanding the underlying processes. In figure 5F it can be seen that MIT's Welkin RDF visualizer easily distinguished the query results as the interplay of 4 collections of 380 *Statements* about 41 *Items* from 5 *Collections* related by 40 *Rules*. For comparison, see Figure 5E where one of its *Statements* is labeled (describing that Age of patient providing pathology sample #90 with Clinical Information #I3646 is 90 years old), along with the parent entities. For examples of other *Statements* about the same *Item* see Fig. 5D. For examples of other statements of the same nature (about the same domain), see 4 statements listed at the bottom-right of Figure 5E.

Conclusion

The Semantic Web[15] technologies have the potential to addresses the need for distributed and evolvable representations that are critical for systems Biology and translational biomedical research. As this technology is incorporated into application development we can expect that both general purpose productivity software and domain specific software installed on our personal computers will become increasingly integrated with the relevant remote resources. In this scenario, the acquisition of a new dataset should automatically trigger the delegation of its analysis. The relevance of this achievement becomes very clear when we note that what prevents a new microarray result from being of immediate use to the experimental Biologist acquiring it is not the computational capability of the experimentalist's machine. Biostatisticians do not

necessarily have more powerful machines than molecular Biologists. Moreover, in neither case is high end computation expected to be performed in the client machine[8]. Rather, once data gathering and data analysis applications become semantically interoperable, at the very least, those who acquire the illustrative microarray data should expect their own machines to automatically trigger its sensible analysis by background subtraction, normalization and basic multivariate exploratory analysis such as dimensionality reduction and clustering. As a consequence, the quantitative scientist's role can be focused on defining the sensibility of alternative contexts of data generation.

The consequences of semantic integration are just as advantageous for those dedicated to data analysis. Statistical analysts typically spend the majority of their time parsing raw datasets rather than assessing the reasonableness of alternative analytical routes. This contrasts with the critical need to validate any given analysis by comparing results produced by alternative configurations applied to independent experimental evidence. It is this final step that ultimately determines the sensibility of the data analysis procedures triggered by the acquisition of data. In summary, any data management and analysis system that will scale for systems level analysis in the Life Sciences has to be semantically interoperable if automated validation is to be attainable.

In this report, we have demonstrated the design of a semantic web data model, S3DBcore, capable of delivering the desired features of distribution and evolvability. This solution relies on RDF triples, the language developed to enable the semantic web in the same fashion that HTML was developed to enable the original web. However, collections of *subject-predicte-object* triples do not establish a management model by themselves. That exercise requires the encapsulation of the data within two additional layers, one confining membership and another permitting access. The effort of identifying management models for information systems has conventionally been the property of technology deployment. This is not feasible when the challenge is scaled to the level of complexity and distribution of Systems Biology. This report describes such a working management model and the authors also make its prototype deployment freely available with open source. In conclusion, a distributed integrated data management and analysis system might look like the prototype infrastructure described in Figure 6 which is based on a semantic database backbone coupled to a code distribution server reacting to the domain of discourse being used.

Author Contributions

Conceived and designed the experiments: HFD CB IW JSA. Performed the experiments: HFD JSA. Analyzed the data: HFD JSA. Contributed reagents/materials/analysis tools: HFD RS DFV CB JDM HRG SGS JAR AMC BB KC AC LHV JSA. Wrote the paper: JSA.

References

1. Blake JA, Bult CJ (2006) Beyond the data deluge: data integration and bio-ontologies. J Biomed Inform 39: 314–320.
2. Komatsoulis GA, Warzel DB, Hartel FW, Shanbhag K, Chilukuri R, et al. (2007) caCORE version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability. J Biomed Inform.
3. Ruttenberg A, Clark T, Bug W, Samwald M, Bodenreider O, et al. (2007) Advancing translational research with the Semantic Web. BMC Bioinformatics 8 Suppl 3: S2.
4. Brazhnik O, Jones JF (2007) Anatomy of data integration. J Biomed Inform 40: 252–269.
5. Hendler J (2003) Communication. Science and the semantic web. Science 299: 520–521.
6. Wiley HS, Michaels GS (2004) Should software hold data hostage? Nat Biotechnol 22: 1037–1038.
7. Wass J (2006) Integrating Knowledge. Bio-IT World 5: 22.
8. Foster I (2005) Service-oriented science. Science 308: 814–817.
9. Hey T, Trefethen AE (2005) Cyberinfrastructure for e-Science. Science 308: 817–821.
10. Nadkarni PM, Miller RA (2007) Service-oriented architecture in medical software: promises and perils. J Am Med Inform Assoc 14: 244–246.
11. Bridges MW (2007) SOA in healthcare, Sharing system resources while enhancing interoperability within and between healthcare organizations with service-oriented architecture. Health Manag Technol 28: 6, 8, 10.
12. Gomadam R, Ramaswamy, Sheth, Verma (2007) A Semantic Framework for Identifying Events in a Service Oriented Architecture. IEEE International Conference on Web Services ICWS 2007: 545–552.
13. Musser J (2006) Web 2.0 Principles and Best Practices; O'Reilly T, ed. O'Reilly Media, Inc.
14. Kamel Boulos MN, Wheeler S (2007) The emerging Web 2.0 social software: an enabling suite of sociable technologies in health and health care education. Health Info Libr J 24: 2–23.
15. Berners-Lee T, Hall W, Hendler J, Shadbolt N, Weitzner DJ (2006) Computer science. Creating a science of the Web. Science 313: 769–771.
16. Berners-Lee T, Hendler J (2001) Publishing on the semantic web. Nature 410: 1023–1024.
17. Gordon PM, Trinh Q, Sensen CW (2007) Semantic Web Service provision: a realistic framework for Bioinformatics programmers. Bioinformatics 23: 1178–1180.
18. Neumann E, Prusak L (2007) Knowledge networks in the age of the Semantic Web. Brief Bioinform 8: 141–149.
19. Post LJ, Roos M, Marshall MS, Driel RV, Breit TM (2007) A semantic web approach applied to integrative bioinformatics experimentation: a biological use case with genomics data. Bioinformatics.
20. Wang X, Gorlitsky R, Almeida JS (2005) From XML to RDF: how semantic web technologies will change the design of 'omic' standards. Nat Biotechnol 23: 1099–1103.
21. Feigenbaum L, Martin S, Roy MN, Szekely B, Yung WC (2007) Boca: an open-source RDF store for building Semantic Web applications. Brief Bioinform 8: 195–200.
22. Borland J (2007) A Smarter Web. Technology Review March/April.
23. Green H (2007) A Web That Thinks Like You. Businessweek 28.
24. Almeida JS, Chen C, Gorlitsky R, Stanislaus R, Aires-de-Sousa M, et al. (2006) Data integration gets 'Sloppy'. Nat Biotechnol 24: 1070–1071.
25. s3db 2.0.
26. Robu I, Robu V, Thirion B (2006) An introduction to the Semantic Web for health sciences librarians. J Med Libr Assoc 94: 198–205.
27. Silva S, Gouveia-Oliveira R, Maretzek A, Carrico J, Gudnason T, et al. (2003) EURISWEB–Web-based epidemiological surveillance of antibiotic-resistant pneumococci in day care centers. BMC Med Inform Decis Mak 3: 9.
28. Stanislaus R, Chen C, Franklin J, Arthur J, Almeida JS (2005) AGML Central: web based gel proteomic infrastructure. Bioinformatics 21: 1754–1757.
29. Ivan Herman RS, DanBrickley (2007) Resource Description Framework (RDF). The World Wide Web Consortium.
30. Pedrioli PG, Eng JK, Hubley R, Vogelzang M, Deutsch EW, et al. (2004) A common open representation of mass spectrometry data and its application to proteomics research. Nat Biotechnol 22: 1459–1466.
31. Orchard S, Jones AR, Stephan C, Binz PA (2007) The HUPO pre-congress Proteomics Standards Initiative workshop. HUPO 5th annual World Congress. Long Beach, CA, USA 28 October-1 November 2006. Proteomics 7: 1006–1008.
32. Orchard S, Montechi-Palazzi L, Deutsch EW, Binz PA, Jones AR, et al. (2007) Five years of progress in the Standardization of Proteomics Data 4(th) Annual Spring Workshop of the HUPO-Proteomics Standards Initiative April 23–25, 2007 Ecole Nationale Superieure (ENS), Lyon, France. Proteomics 7: 3436–3440.
33. Klimek J, Eddes JS, Hohmann L, Jackson J, Peterson A, et al. (2007) The Standard Protein Mix Database: A Diverse Data Set To Assist in the Production of Improved Peptide and Protein Identification Software Tools. J Proteome Res.
34. Aho JDU AV (1979) Universality of data retrieval languages. Proceedings of the 6th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. pp 110–119.
35. Aranguren ME, Bechhofer S, Lord P, Sattler U, Stevens R (2007) Understanding and using the meaning of statements in a bio-ontology: recasting the Gene Ontology in OWL. BMC Bioinformatics 8: 57.
36. Lam HY, Marenco L, Shepherd GM, Miller PL, Cheung KH (2006) Using web ontology language to integrate heterogeneous databases in the neurosciences. AMIA Annu Symp Proc. pp 464–468.
37. Zhang S, Bodenreider O, Golbreich C (2006) Experience in reasoning with the foundational model of anatomy in OWL DL. Pac Symp Biocomput. pp 200–211.
38. Miller M, Rifaieh R (2006) Wrestling with SUMO and bio-ontologies. Nat Biotechnol 24: 22–23; author reply 23.
39. Musen MA, Lewis S, Smith B (2006) Wrestling with SUMO and bio-ontologies. Nat Biotechnol 24: 21; author reply 23.
40. Stoeckert C, Ball C, Brazma A, Brinkman R, Causton H, et al. (2006) Wrestling with SUMO and bio-ontologies. Nat Biotechnol 24: 21–22; author reply 23.
41. Blake J (2004) Bio-ontologies-fast and furious. Nat Biotechnol 22: 773–774.
42. Soldatova LN, King RD (2005) Are the current ontologies in biology good ontologies? Nat Biotechnol 23: 1095–1098.

43. Merelli E, Armano G, Cannata N, Corradini F, d'Inverno M, et al. (2007) Agents in bioinformatics, computational and systems biology. Brief Bioinform 8: 45–59.
44. Antoine Isaac JP, DanielRubin (2007) SKOS Use Cases and Requirements.
45. The University of Texas Lung Cancer SPORE. P50 CA70907.
46. Baker T (2005) A Common Grammar for Diverse Vocabularies: The Abstract Model for Dublin Core. Lecture Notes in Computer Science 3815: 495.
47. Good BM, Wilkinson MD (2006) The Life Sciences Semantic Web is full of creeps! Brief Bioinform 7: 275–286.
48. Neumann E (2005) A life science Semantic Web: are we there yet? Sci STKE 2005: pe22.

Almeida et al. BMC Bioinformatics 2010, 11:387
http://www.biomedcentral.com/1471-2105/11/387

BMC Bioinformatics

RESEARCH ARTICLE Open Access

S3DB core: a framework for RDF generation and management in bioinformatics infrastructures

Jonas S Almeida[1*], Helena F Deus[1,2], Wolfgang Maass[3]

Abstract

Background: Biomedical research is set to greatly benefit from the use of semantic web technologies in the design of computational infrastructure. However, beyond well defined research initiatives, substantial issues of data heterogeneity, source distribution, and privacy currently stand in the way towards the personalization of Medicine.

Results: A computational framework for bioinformatic infrastructure was designed to deal with the heterogeneous data sources and the sensitive mixture of public and private data that characterizes the biomedical domain. This framework consists of a logical model build with semantic web tools, coupled with a Markov process that propagates user operator states. An accompanying open source prototype was developed to meet a series of applications that range from collaborative multi-institution data acquisition efforts to data analysis applications that need to quickly traverse complex data structures. This report describes the two abstractions underlying the S3DB-based infrastructure, logical and numerical, and discusses its generality beyond the immediate confines of existing implementations.

Conclusions: The emergence of the "web as a computer" requires a formal model for the different functionalities involved in reading and writing to it. The S3DB core model proposed was found to address the design criteria of biomedical computational infrastructure, such as those supporting large scale multi-investigator research, clinical trials, and molecular epidemiology.

Background

The increasing adoption of semantic web technologies and formalisms in biomedical and biomolecular areas is often driven by the need to interoperate between ever more complex data stores and between the applications that process them [1-3]. As the pace of adoption quickens, a distributed infrastructure is emerging that is starting to satisfy the two properties of a von Neumann architecture, also known as "stored-program computer": that it can store both data and the applications. The mingling of data and ready to run applications is particularly tightly woven in web services that rely on padded JSON calls (JSON: Java Script Object Notation), following on proposals for crossdomain JSON calls such as [4], and are now used by major web 2.0 services [5,6]. In those systems, if properly configured [7], there are no syntactic barriers to workflows that pull data and code from different machines and then transfer the results to further use elsewhere.

By breaching the same origination restriction of URL calls in the conventional (XML based) AJAX model, JSON-based systems also subvert the client-server equation. Specifically, JSON calls are function calls where the data is passed as the input argument and the function name is specified as a callback parameter in the URL call. This signifies that data and code can be invoked freely from multiple originations and can be made part of arbitrary workflows, as in the node.js project [8]. This outcome, illustrated in Figure 1, was also anticipated by the view of the semantic web as leading to an ecosystem of usages accessing a shared "RDF-bus" [9]. It also suggests an architecture for distributed bioinformatics infrastructures that literally delivers the "web as a computer", that is, a von Neumann machine. Indeed, the work described here can be construed as an attempt to build the minimal set of server-side features that facilitate the integrated management of data and of its analysis in a web infrastructure.

* Correspondence: jalmeida@mathbiol.org
[1]Department of Bioinformatics and Computational Biology, The University of Texas M D Anderson Cancer Center, 1515 Holcombe Blvd Houston, TX 77030, USA

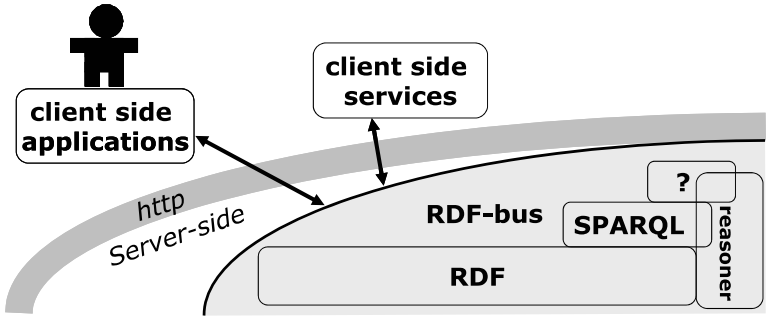


Figure 1 Web-based infrastructure architecture composed of server side representation and client side presentation + data analysis computational services. This disposition moves to the client side both the assembly of interfaces as well as the computational intensive data analysis services - such as computational statistics modules. As a consequence, all server side components are standardized and can therefore benefit from cloud computing scaling.

In a strict sense, a von Neumann architecture comes with a von Neumann bottleneck [10] in data access. However, the client side applications described in Figure 1 are hosted in independent machines, that is, in machines with their own memory where data can be cached for ready access by the CPU. Therefore this architecture is more accurately described as a von Neumann hybrid supporting Non-Uniform Memory Access (NUMA). In a nutshell, the distributed computing enabled by web-like architectures have fundamental advantages for scalability that stem from the memory access architecture and the reliance on functional programming (JavaScript) along lines anticipated by John Backus 1977 Turing Lecture, which are now key to data-intensive scientific discovery [11,12].

The use of server-side only as a standardized representation layer for scientific research applications is not original. It is, for example, at the core of cloud computing based systems such as Google Wave [13]. In such systems, the computational intensive data processing components can be deployed as client-side services that regularly consult the representation of the domain they were written to process. In the illustrative Google Wave example those client-side services are designated as "Robots" [14]. What is originally proposed here, and is illustrated with prototype applications, are minimal abstractions that will support the requirements of a distributed bioinformatics infrastructure.

Design Criteria

The architecture described in this report specifically targets Biomedical applications, which places two requirements on a web-based infrastructure. First, it needs to accommodate the fluidity that is intrinsic to the Biology knowledge domain [15]. Second, individual variability is driving the redesign of biomedical information management systems to allow mixing private and public data in personalized medicine applications [16,17].

The first design criterion of flexible handling of fluid and heterogeneous domains suggests that the Biology domain expert should control the description of the domain that is being experimentally explored. In fact, the results of data analysis often require the redesign of the original data acquisition effort. The redesign happens so patently that, for example, it is now explicitly exploited to speed up target identification and drug discovery through the use of adaptive designs in clinical trials [18]. Redesign is also often triggered in Biomedical research by advances in the analytical methods and can become the major challenge to the use of new technologies, as is currently the case for next generation sequencing [19]. In either case that redefinition of domain has to be achieved without compromising the consistency of the data already acquired. Another consequence of this domain fluidity is the need for co-existence of a myriad of sub-fields and subcommunities which are not necessarily in agreement with each other. This property alone suggests that bioinformatics infrastructure should support bottom-up, collaborative, data acquisition and representation linked to multiple descriptions of the same domain. The motivation for this design criterion is therefore the accommodation of the widest range of data acquisition efforts in the same web-based infrastructure. The resulting resource would be useful as the starting point for the identification of logical models, while not being constrained by them.

The second design criterion, that of fine grained management of access permission, calls for a generic mechanism to describe the relationship between the user and each data element. That description could then be used by the infrastructure to decide what types of access to the data are authorized for each user. A literal

reading of this requirement would be to document that relationship for each element, individually, and for each user. This absurd solution would of course increase the size of the data repository several fold. A more scalable alternative is therefore needed which allows for that relationship to be also defined for the description of the domain, and then propagated to its observational instantiation. Accordingly, the identification of a Markov model that propagates user relationships among the S3DB entities is the second key feature of the core model described in this report.

Methods

The abstractions described in the Results resort to W3C formalisms, are illustrated by an accompanying library and are partially deployed by a web-service:

Core model entities

The description of the core entities of the S3DB core model was pursued with recourse to the World Wide Web's Consortium (W3C) Resource Description Framework (RDF) [20], including related schema language RDFS [21], and OWL Web Ontology Language [22].

Propagation of user operators

S3DB operators describe relationships between users and entities of the core model. Any operator predicated on a user as a subject, and on any of the seven S3DB core entities as an object, will be propagated as a Markov process. The propagation model described in the results section was originally coded as a finite state automata (FSA) using MATLAB (Mathworks Inc) and consists of three functions: merge, migrate and propagate. These m-functions were written to be also executable in less sophisticated open source m-code interpreters such as freemat http://freemat.sourceforge.net. The three functions were then also coded in javascript to support web browser based applications such that their inner workings can be explored without need for specialized programming environments. These applications, with links to the m and js source code, are made available at http://s3db-operator.googlecode.com. The Markovian process described by them was used in the prototype web service (also freely provided with open source, see next section) to calculate the independent propagation of each of the three permission operators supported by that particular implementation - *View*, *Edit* and *Use* - for each of the three permission states considered - *none*, *self* and *all*.

Web service prototype

The identification of the S3DB core model has been pursued for five years as an iterative exercise where tentative new features in the core model were exposed to communities of biomedical and molecular epidemiologists to collect usage feedback [15,23-27]. This feedback typically came with suggestions for desirable behaviors that informed the next round of core model re-design. A regularly updated version of this prototype web service is available for download with open source at the http://s3db.org project web site and also at http://s3db.googlecode.com. The webservice's API is exposed through a REST protocol, S3QL, documented at http://s3db.org/documentation/s3qlsyntax. A javascript library for cross domain JSON requests is also provided at http://s3dbcall.googlecode.com.

Results

The advantages of a "sloppy", evolvable, data representation distinguishing between domain and instantiation was first argued in [15] using a relational diagram. That argument was expanded and a first draft of the core model was subsequently used to integrate distributed data sources for a Lung Cancer SPORE [24], and to enable the realtime analysis of DNA copy number variation (CNV) in glioblastoma multiforme tumor samples [25], and to support a standards based proteomic repository [26]. A complete model, designated as "s3db core model", needs to merge that draft logical model with the Markovian propagation of user operators used to assign user permissions to S3DB entries.

Separating Domain from its observational instantiation

The separation of domain from instantiation is centered on the pattern described in lower half of Figure 2, where the representation of domain as triples is shown to be the predicate of the statements that instantiate that domain with observations. The 7 core entities and 12 logic relationships between them, outlined graphically in Figure 2, are more formally described in Table 1. **In a nutshell, the use of the S3DB model ultimately consists of declaring all the data elements associated with a given observation as being types of S3DB entities**.

The core entities

The root entity of each S3DB representation is the Deployment, which is identified by the location of the S3DB service. A Deployment is directly related (first order) with Users and Projects (*s3db:DP* and *s3db:DU* in Figure 2 and Table 1), with the latter providing granularity for sets of Collections and Rules (*s3db:PC* and *s3db:PR*). The Collections are used as subjects and as objects of Rules, which represent the domain one wants to instantiate with observations (*s3db:Rsubject* and *s3db:Robject*). For example the concept that "people live in places" would be represented by a Rule associating the Collection of people with the Collection of locations (see diagram at the bottom of Table 1, and relationships 5-10 in Table 1). The Collections in turn delimit sets of

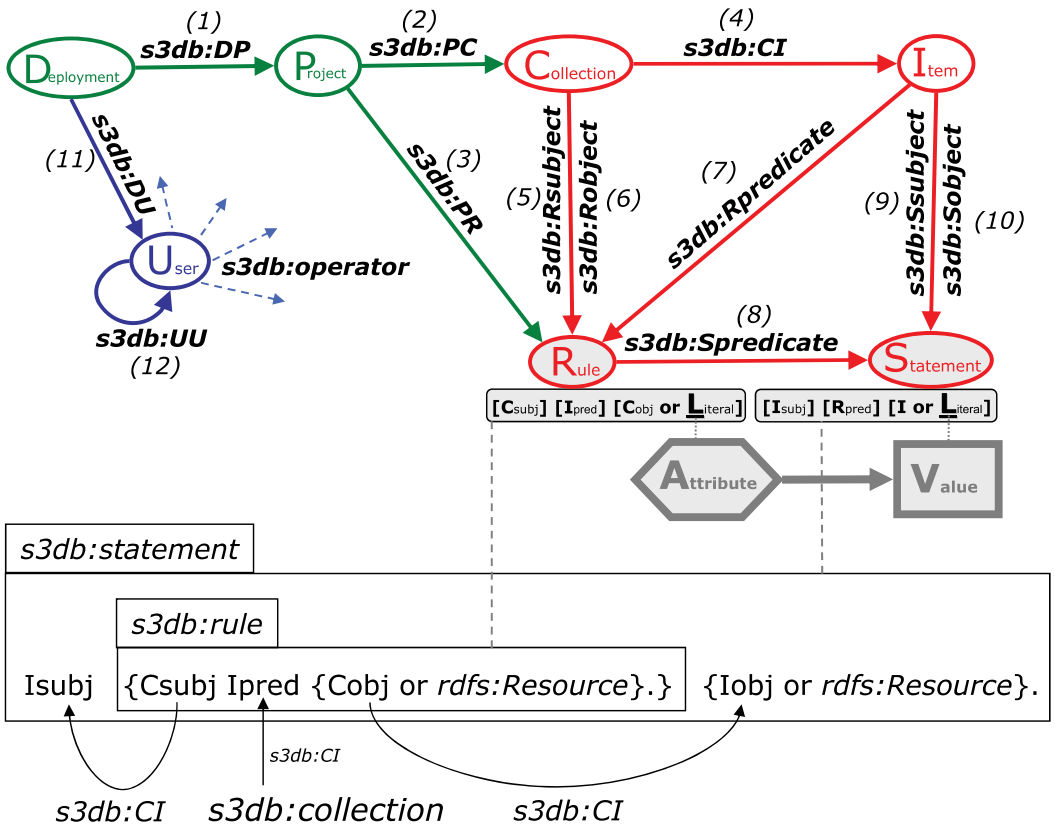


Figure 2 Two views of the S3db core model. Top diagram - solid arrows describe relationship between the seven core entities; Dashed arrows (s3db:operator) indicate operators whose states describe the relationship between users and each of the core entities. Bottom box diagram - detail on the key relationship between s3db:rule and s3db:statement using N3 notation. The s3db:rule is a dyadic predicate and it is also, as a whole, the predicate of the s3db:statement. If the object of the s3db:rule triple is not a Collection, then the object of the Statement that rule predicates will be the attribute's literal value. Otherwise the statement's object is the item of the collection indicated as object of the rule. The statement subject is invariably an item from the collection indicated as subject of the predicate rule. See Table 1 for nomenclature and definitions. In the reference prototype (available at s3db.org), the former situation is indeed handled as a literal in a *varchar* database field, as noted in the diagram on top as corresponding to the Attribute/Value model nuclei. However, what that signifies as regards the underlying model is that these two objects can be anything, as are therefore more accurately noted as a rdfs:Resource in the box diagram.

Items (*s3db:CI*). For example, *Mary* could be an Item of the Collection of people. In some cases the object of the rule is best not confined by a Collection and instead can be left as a Literal, as in, for example, the instantiation of the Rule "people have names". Note no cardinality is imposed for any relationship so many-to-many scenarios are allowed. For example, the same Item can be a member, *s3db:CI*, with multiple Collections. On the contrary, we found it useful to restrict the predicates of the Rules, such as *live in* and *have* above, to be Items of Collections (see *s3db:Rpredicate*, #7 in Table 1), as will be clearer after describing the propagation of user operators.

Data submission to a S3DB service corresponds to instantiating those rules with the observations, through the use of Statements. For example, the Statement "Mary lives in Houston" would have a rule like "people live in places" as a predicate associating an Item of the Collection of people (*Mary*) with an Item (*Houston*) of the Collection *place*. One could then continue weaving the description with further assertions by first identifying new components of the domain, for example, by creating a Rule to the effect that "places have addresses" and then asserting those addresses as Items of the Collection of places. It is important to recall that each non-literal element is identified by a Universal Resource

Table 1 Minimal description of the core 12 relationships and 1 operator between the 7 s3db entities, using notation 3 (N3)

(s3db:deployment s3db:project s3db:collection s3db:item s3db:rule s3db:statement s3db:user) rdfs:subClassOf s3db:entity.

(s3db:DP s3db:PC s3db:PR s3db:CI s3db:CI s3db:Rsubject s3db:Robject s3db:Rpredicate s3db:Ssubject s3db:Sobject s3db:Spredicate) rdfs:subClassOf s3db:relationship.

1. *s3db:DP rdfs:domain s3db:deployment; rdfs:range s3db:project.*

2. *s3db:PC rdfs:domain s3db:project; rdfs:range s3db:collection.*

3. *s3db:PR rdfs:domain s3db:project; rdfs:range s3db:rule.*

4. *s3db:CI rdfs:domain s3db:collection; rdfs:range s3db:item.*

5. *s3db:Rsubject owl:inverseOf rdf:subject; rdfs:domain s3db:collection; rdfs:range s3db:rule.*

6. *s3db:Robject owl:inverseOf rdf:object; rdfs:domain s3db:collection; rdfs:range s3db:rule.*

7. *s3db:Rpredicate owl:inverseOf rdf:predicate; rdfs:domain s3db:item; rdfs:range s3db:rule.*

8. *s3db:Spredicate owl:inverseOf rdf:predicate; rdfs:domain s3db:rule; rdfs:range s3db:statement.*

9. *s3db:Ssubject owl:inverseOf rdf:subject; rdfs:domain s3db:item; rdfs:range s3db:statement.*

10. *s3db:Sobject owl:inverseOf rdf:object; rdfs:domain s3db:item; rdfs:range s3db:statement.*

11. *s3db:DU rdfs:domain s3db:deployment; rdfs:range s3db:user.*

12. *s3db:UU rdfs:domain s3db:user; rdfs:range s3db:user.*

s3db:user s3db:operator s3db:entity.

All relationships except for s3db:operator (last row) are s3db:relationship (first row). The inversion of RDF subject, predicate and object in relations 5-10 may appear capricious at this point but it will simplify the identification of automata for the propagation of s3db:operator states in the next section. Specifically, it will allow the definition of Equation 3 such that the direction of the arrows in Figure 2 is the same as the propagation of s3db:operator states.

Identifier (URI), necessary to make assertions using RDF. In conclusion, the purpose of the S3DB core model is just to provide a template where to aggregate data elements that may already be available, either as their own pre-existing URIs, or, otherwise, by generating those URIs within S3DB.

The illustration of the previous paragraph will now be repeated using the formalism of notation 3 (n3) and the RDF, RDFS and OWL vocabularies (see Methods), with reference to the list of 12 relationships described in Table 1.

We can now return to the example that "Mary lives in Houston" and use the S3DB template to generate the triples to be submitted to the S3DB service. Starting with a deployment hosting an instance of a s3db:project, *P_example*, and using notation 3 (N3),

a) Create Collections of people and places:

:P_example s3db:PC :C_person.

:P_example s3db:PC :C_places.

b) Insert Mary and Houston as items of the respective collections:

:C_person s3db:CI :I_Mary.

:C_places s3db:CI :I_Houston.

c) Describe the domain we are about to instantiate, that people live in places, as a

s3db:rule:

:P_example s3db:PR :R_people_in_places.

:C_person s3db:Rsubject :R_people_in_places.

:C_place s3db:Robject :R_people_in_places.

:I_lives_in s3db:Rpredicate :R_people_in_places.

(lets not worry about what collection of Items *I_lives_in* comes from just yet).

d) Insert the new data:

:I_Mary :R_people_in_places :Houston.

This example illustrates a very simple mechanism to store descriptions of the domain and the data that instantiates them in such as way that they can be edited as required by the fluidity of the life sciences domain [15]. The actual identifiers, such as *":I_Mary"*, *":C_ person"* or *":P_example"* in reality are random or sequential alphanumeric strings such as the unique indexes generated by the S3DB service. Of course *: I_Mary* has a name, which we will use as an example to illustrate how literal values are asserted through instantiation of a Rule, without the need for, say, mediation by a Collection of names:

@prefix foaf: <http://xmlns.com/foaf/0.1/>.

:P_example s3db:PR :R_people_have_ names.

:C_person s3db:Rsubject :R_people_have_names.

foaf:firstName s3db:Robject :R_people_have_names.

:I_has s3db:Rpredicate :R_people_have_names.

which then allows inserting the corresponding literal information,

:I_Mary :R_people_have_names "Mary".

Note in this last assertion that neither the object of the Rule, *foaf:firstName*, nor the object of the Statement, "Mary", are s3db entities. As noted in Figure 2 and discussed in its legend, when the object of the Rule is not a Collection, the core model allows for any type of content (any other type of *rdfs:Resource*) to be associated to either object in the Rule/Statement instantiation. In the implementation followed by the reference S3DB prototype, these two non-s3db entities are simply stored as literals in a variable length string (*varchar*) database field.

More importantly than the data type of the Rule and Statement objects is that if Mary changes her name that doesn't affect the information about where she lives or that she has a first name. The reverse is also true, if what she has is no longer designated *foaf:firstName*, it could even be replaced by another literal such as "name", that editing does not affect the existing assertion: whatever the new designation is, it is still instantiated in the same Statement, with the same URI, by the same *rdfs:Resource*, in this case the literal "Mary". This is far from being an esoteric scenario. In molecular epidemiology surveillance it is quite common to have, for

example, the identity of a microbial isolate, which is an instance of a class, be used as the predicate of the molecular typing methodology. There is of course nothing new here; this modularity is intrinsic to the dyadic predicated nature of the RDF framework. However, what was achieved by tying the submission of new data to the S3DB core model was to restrict the use of RDF such that an explicit distinction of domain and observational instantiation is preserved throughout the process. As will be discussed later, this was achieved purely through the *assertion* [28] of a design pattern, that is, without the computational overhead of description logics and the need of reasoners for subsequent information retrieval.

Propagation of S3DB operator states

The last relationship in Table 1, *s3db:operator*, is the point of entry for the second component of the core model, the embedded finite state automata (FSA) that propagates the states of any such operator. This component allows the assertion of a generic relationship between a user and a component of the domain, for example, Collections and Rules, and then expect to find it automatically propagated for its instantiation, for example, as Items and Statements (Figure 2). Inversely, S3DB operators can also be used to define exceptions to broader relationships, for example, by using operator states to describe relationships between the User and individual Statements, or Items, without affecting the remainder entries in the same set, that is, for other Statements on the same Rule or Items of the same Collection. This model was identified in very generic terms as to allow for the definition of complex relationships to be described succinctly, without the need to particularize them for each entry. Throughout the different projects where S3DB was used, we have found the critical need for a solution that is balanced between the two extreme scenarios of having a system where all user permissions are indiscriminately set at the point of access, and the extreme alternative of having user group permissions for every entry. Accordingly, the embedded propagation of user relationships was first devised narrowly as a solution for the challenges of mixing public and private data, as well as mixing data instantiating distinct, even contradictory, descriptions of a domain in multiple investigator initiatives. It was only in the later stages of the project that the opportunity for a generic solution of propagating unspecified operator states became apparent.

An s3db operator, f, is a discrete variable with a set of n ordered states. The elements of the set can exist in two different forms, a upper case or dominant form, Φ, and a lower case or recessive form, φ. For example, the capital form of the i^{th} state of the operator f, would be represented as Φ_i, where $i \in [1,..., n]$. Accordingly, the description of such relationship between a user and some *s3db:entity* is defined using the state of the operator, as represented in Equation 1:

$$\begin{array}{lll} f & subClassOf & s3db:operator. \\ (\phi_i, \Phi_i) & subClassOf & f. \\ U\_some\_user & (\phi_i, \Phi_i) & E\_some\_entity. \end{array} \quad (1)$$

The three functions described below, merge, migrate and percolate, are used in the resolution of state propagation between data elements. That description is best followed by testing different scenarios using the accompanying tool at http://s3db-operator.googlecode.com.

Merging

As illustrated in Equation 1, for each user, U, and for each instance, E, of any of the seven types of *s3db:entity*, the nature of the relationship can be described by an arbitrary number of states of the operator f, by simply declaring the $\{U\ f\ E\}$ triple. However, regardless of such statements having been made between a User and an Entity, the f state assigned as predicate in those statements is not necessarily the effective state of that relationship. Other states may also be indirectly asserted to the relationship by directly assigning them for relationships with entities upstream of the target entity. The resolution of what state is effective for the relationship between a given U and E is resolved by merging all the assigned states, directly or indirectly, as defined in Equation 2. In this equation, A is the vector of indexes of assigned dominant (upper case) states,Φ; and a is the index vector of assigned recessive (lower case) states, φ. As for the other definitions, the behaviour and implementation of merge can be verified using the accompanying tool at http://s3db-operator.googlecode.com.

$$i = merge(\{\Phi_A, \phi_a\}) \rightarrow \begin{cases} i_{|A=null} = \max(a) \\ i_{|A\neq null} = \min(A) \end{cases} \quad (2)$$

The numeric indexes of the vectors A and a, are integers between 1 and n. However, because numbers are symbols with no upper and lower case, it is easier to represent the resolution of Equation 2 using the alphabetic indexes instead. The argument for using alphabetic indexes is that their case can distinguish between a dominant and a recessive merged state, therefore allowing a and A to be represented together as a single vector. Two illustrative examples - for an operator with three states indexed as {'b','c','d'}, merge({'b','c','d'}) = merge('d') = 3 and merge({'b','c','C','D'}) = merge('C') = 2. The case of the merged state, 'd' and 'C' in the example, is of no consequence to the operator itself, which will respond only to its position in the ordered state

vector, 2 and 3 respectively. If, in this example, the operator was something like *view_query_results()* and the index of the ordered states were {'noView','theCountOnly','yes'}, the result of the first merging might be returned as 'yes' and of the second as view 'theCountOnly'. However, if further operations are to be made on the merged result then the case of the merged state is important and needs to be retained. It is patently easier to return 'd' and 'B', or even 'yes' and 'THECOUNTONLY' than to have to specify that the merged i = 3 was a recessive outcome, whereas i = 2 was dominant, and as a result the state index 2 > 3 when Equation 2 is used.

Migration

The direction of the relationships between S3DB entities (Figure 2, Table 1) was conveniently defined to be the same as the propagation of operator states from domain to its instantiation (note inversion of *rdf:subject*, *rdf:predicate* and *rdf:object* in relationships 5-10, Table 1). This allows the definition of a Boolean transition matrix, Equation 3, that can be applied to any instance of one or more of the seven types of s3db Entity, *E*, ordered using their initials as *[D, P, C, R, I, S, U]*. The numbers between brackets in the transition matrix indicate the logical tests (as numbered in Table 1) that individual instances of the seven types of entities can have between each other (Figure 2).

$$T_{S3DB} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ (1) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & (2) & 0 & 0 & 0 & 0 & 0 \\ 0 & (3) & [(5),(6)] & 0 & (7) & 0 & 0 \\ 0 & 0 & (4) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & (8) & [(9),(10)] & 0 & 0 \\ (11) & 0 & 0 & 0 & 0 & 0 & (12) \end{bmatrix}, E = \begin{bmatrix} D \\ P \\ C \\ R \\ I \\ S \\ U \end{bmatrix} \quad (3)$$

As described in Equation 4, this simplifies the computation of the transition of states between entities as the external product of the corresponding Boolean square matrix and the vertical vector of states assigned to each entity. For example, if a state of a *s3db:operator* is used to describe a User relation with a certain *s3db:collection*, and this Collection happens to both have Items and to be the Subject of a Rule, then this state will be passed to those Rules and to those Items, using relationships (4) and (5), respectively.

$$E_{k+1} = merge\left(\left[E_k, migrate(T \times E_k)\right]\right) \quad (4)$$

The process by which states are passed from one instance of an entity to another before being merged at the end of each iteration (Equation 4) is designated as state migration and is described in Equation 5. The simplest example is the migration of a singular state - if the state of an instance of a *s3db:entity*, *E*, is described as a singular value, say 'a', then 'a' will be passed on for the relationships verified in Equation 4. However, if the state of the operator, *f*, is described by more than one value, *l*>1, then the additional expressivity in state propagation can be achieved, as described by Equation 5. That generalization consists of specifying that if a state is singular ($l = 1$), then it will be passed as is. If, on the other hand, it is plural, then the first state is used as the effective state of the subject entity and only the remaining states are passed on to the entity that is object of the valid relationship, as described in Equation 5. For example, starting with singular migration, if the state of an instance is 'a', and this instance is subject of one or more of the 12 relationships (Table 1), then the object state will merged with the migrated state 'a'. However, if the subject instance has a plural state, say 'abCd', then only 'bCd' will migrate. Note that both dominant and recessive cases are considered in the vector *f* in Equation 5.

$$\begin{aligned} &f_{object,k+1} = merge([f_{object,k}, migrate(T \times f_{subject,k})]) \\ &l = length(f) \\ &l = 1 \rightarrow migrate(f) = f = f[1] \\ &l > 1 \rightarrow migrate(f) = f[2, \ldots, l] \end{aligned} \quad (5)$$

One last generalization of the migration process was also found to increase expressivity. The procedure described in Equation 5 was vectorized to allow simultaneous migration of states of multiple s3db operators. This is achieved by defining a second input argument for the *migrate* procedure which identifies how many operators f_j, $j = 1,\ldots,m$, are having their states migrated simultaneously. Since the states of each operator define m-tupples inside the state of *n* states, this is equivalent to identifying the migrating states of f_j as being $f_j = f[i, i+m, i+2m, i+3m, \ldots, n*m]$. Accordingly, Equation 6 is equal to Equation 5 when m = 1, that is, when only one operator is being considered.

$$\begin{aligned} &i = [1 + m, \ldots, \max([2m, m \cdot ceil(l / m)])] \\ &i > l, i - m > 0 \rightarrow f[i] = f[i - m] \\ &i > l, i - m \leq 0 \rightarrow f[i] = f[i - 1] \\ &migrate(f, m) = f[i] \end{aligned} \quad (6)$$

The enhanced expressiveness of the representation of multiple operator states described in Equation 6 is most useful for s3db operators that share the same states. For example, states that identify groups of users could be used as the states of multiple operators such as "view" and "edit", as is the case for the S3DB prototype (see Methods). As can be verified in the tool accompanying this manuscript, the multiple state migration allows for very short descriptions of states that span multiple

operators. For example, migrate('a',3) = 'aaa', which allows for a single assignment that spans several operators. This is achieved without affecting the migration of individual statements - for example migrate('abc',3) = 'abc'- while at the same time allowing for a sweeping assignment of migrated states as in migrate('abcb',3) = 'bbb'. Note also in Equation 6 that when a operator state at position i is not specified, it is borrowed from the operator immediately to the left, position i-1. This implies that the order of the operators can be used to simplify assignments that just span a subset of them, as in migrate('abcbc',3) being 'bcc'. As always, the behavior and implementation of this functionality can be verified using the accompanying tool.

Percolate

The third and last function used by the state propagation procedure brings together the merge and migration functions to find the steady state solution of Equation 4. That is, when the migration of states, Equation 6, has progressed to the point where the effective state of the operator, for each and every s3db entity, no longer changes:

$$E_{k+1} = E_k \tag{7}$$

In the accompanying web tool this resolution is made available for any Boolean transition matrix. Although for the specific purposes of the S3DB prototype, the transition matrix T in Equation 4 is equal to *TS3DB* in Equation 3, there is no reason not to define, and test using the tool, the percolation of s3db operator states more broadly for arbitrary transitions.

Discussion

The S3DB framework comprises a core model with an embedded Markov process propagating user operator states. The two key properties of the resulting construct are the explicit separation between domain and its experimental instantiation, and the accommodation of a very flexible and fine tuned description of the relationship between the users and its contents. Most features of this framework were put to use in an open source prototype available at s3db.org. They have also been validated with practical applications developing multiple investigator information management infrastructure such as [24]. The potential usages and configurations of the S3DB framework described here are nevertheless much broader and can be described as a set of restrictions placed on RDF representations. Recalling from the Results section, **the use of the S3DB model consists of declaring all the data elements associated with a given observation as being types of S3DB entities**. The interoperability with the resulting construct is ideally delivered as a REST web service protocol, which is not covered in this report. For an illustrative implementation see the prototype's documentation for the query language S3QL, to which SPARQL queries can also be mapped http://sparql.s3db.org.

Core model

As described in the box diagram at the bottom of Figure 2, the key feature of the core model is the representation of value triple statements (*s3db:statements*) predicated on triple statements that describe the domain instantiated (*s3db:rules*). This design pattern was specifically devised to allow domain experts to incubate the description of their own domain of expertise [15]. As proposed in that report, and verified here, that pattern establishes a specific relationship between the Objects and Subjects between the two triples, which allows the autonomous editing of the domain description and of its value instantiation.

The three core s3db entities peripheral to the nuclear square of Collections, Items, Statements and Rules (in red in Figure 2) create additional management modeling opportunities without which the core model would be little more than a flexible data format. The first of these three entities is the Deployment, which was conceived as a pointer to the URL of the S3DB web service. As all model entities, its usage is not conditioned, nor does it condition, the location of the other entities (Users, Projects and Rules in this case) linked to it. This design seeks to support the distribution of the information management infrastructure. This can be achieved for example, by dereferencing. The Deployment URL address can point to a central registry that resolves it to the actual web service. In that case, the content hosted can be distributed between multiple machines for purposes of, say, load balancing, or in general to enable intermediate tools that may aid in content discovery. Another example is the compartmentalization of content by hosting Collections and Rules in multiple machines, distinct from the hosting Projects (in green in Figure 2). Specifically, because the project URI is resolved by the hosting deployment, the relationship *s3db:DP* (Table 1) can be used to associate deployments with arbitrary Projects, which do not have to be in the same machine that hosts the subject Deployment. Furthermore, by defining URI's for individual entities as URL calls to the Deployment this discussion is extensible to all other relationships in Table 1. This can also be verified by following this link [29] to a S3DB project (click Enter on the key field to login as a public user), with content retrieved from The Cancer Genome Atlas (TCGA). Note the resolved URI's at the lower left corner of the web application.

The last of the four peripheral Entities is the User which is *rdf:subject* of a class of operators with states

that were conceived for fine grained definition of relationship between users and the corresponding content. As discussed above for Deployments, Collections and Rules, the URI of a user can also be resolved to any deployment by *s3db:DU* (#11 in Table 1). This implies that authentication and user management are architecturally decoupled. The users have also specified a relationship between themselves, *s3db:UU* (#12 in Table 1), conceived to support a flexible definition of grouping. When two users are linked by *s3db:UU*, the operator states can migrate between them to extend or restrict the relationships with the corresponding content. This enables the use of *s3db:UU* to design flexible user management systems. For example, if one User entity is used as a hub to which multiple UU relationships converge, this is akin to the conventional definition of groups. The reverse would be closer to the conventional definition of roles. A rich variety of groupings can be envisioned between these two conventional extremes. Note that contrary to the operator percolation features discussed in the two previous paragraphs, between-user percolation is presently not supported, and therefore was not tested, by the reference S3DB prototype. Two ongoing projects in particular, http://aguia.googlecode.com and http://cnviewer.googlecode.com, provide S3DB-based, javascript-coded, platforms where those features are being explored, respectively, in the context of clinical trial management and of DNA copy number variation in tumor samples.

S3DB operators

The idea of s3db operators as a class of functions with states that describe the relationship between a user and the entities of a data model was conceived independently from its specific application to the S3DB core model. Accordingly, this second component of the core model is applicable to any RDF schema provided that the direction of state propagation is defined (or in its absence by the directionality of RDFS/OWL model, as is the case of the S3DB schema). The generic nature of the S3DB user operators and of the Markov process that propagates its states is apparent in Equations 2,4-7. Similarly, although the S3DB prototype provides an illustrative example using three S3DB operators ('view', 'edit' and 'use'), each with the same three ordered states ('yes', 'self', and 'no'), the range of applications is open ended and is not necessarily associated with permission management. For example, an operator could be defined to represent priority in the retrieval of query results, which could differ between users. By using the states of this operator to define the relationship between a user and the model entities one could configure, for that user, that clinical records with a specific outcome would be, for instance, graphically highlighted. Retrieval priorities, or, for that matter, the choice of graphic interface features, could therefore be personalized by associating them to operator states pointed to the appropriate semantic content.

The graphic presentation beyond access control could also include workflow components. For example, it could be used by a quantitatively minded researcher to have statistics tools automatically applied to the construction of a specialized interface. More interestingly, the concept of User could be used more broadly as that of usage. Quite literally, data analysis procedures could be configured as Users. By treating usages and analytical workflows as users, the corresponding procedure would be automatically executed for content with specified semantics. The same line of discussion can also lead to the observation that there is nothing in the definition of an *s3db:operator* that restricts its use to describe the relationship between a user and the entities of the s3db core model. That restriction is described by the construction of the transition matrix which for this core model happens to be the one defined in Equation 3. Therefore, a different core model would just have to identify a different transition matrix of logical tests and the same state propagation mechanism defined in Equations 2,4-7 would be automatically applicable. In summary, the key feature of the user relation propagation component of the S3DB core model is the articulation of the three functions, merge, migrate and percolate, applied to a set of states that can take a dominant or recessive form. That articulation is defined by those equations and is also illustrated by the accompanying browser-based application at http://s3db-operator.googlecode.com.

Conclusion

The dyadic predicated nature of Resource Description Framework (RDF) has emerged as the shared representation of a variety of semantic web formalisms and technologies. In this report we describe a core model that mediates the generation and management of the RDF triples by and for domain experts. This model abstraction is the result of five years of bioinformatics infrastructure development in biomedical and molecular epidemiological contexts. The underlying approach/hypothesis is that by explicitly distinguishing description of the domain from its instantiation with observational data, one allows domain experts to freely evolve the former without compromising the actuality of the latter. The other, complementary, critical feature of the S3DB core model is the Markov process that propagates the relationship between users and the entities of a core model. This ability to propagate operators from the description of the domain to its instantiation has already found an immediate application in the management of

access permissions. Finally, at the very center of the S3DB abstraction is a two tiered modeling pattern where instances of a class describe the relationship between two classes and, in turn, instances of the resulting triple, which are also triples, host the observed values. This modeling pattern underlies the S3DB schema but may be of more general applicability.

Acknowledgements
This work was supported in part by the Center for Clinical and Translational Sciences of the Texas Medical Center at Houston under NIH (CTSA) contract no. 1UL1RR024148, by the National Cancer Institute grant 1U24CA143883-01, by the European Union FP7 PNEUMOPATH project award and by the Portuguese Science and Technology foundation under contracts PTDC/EIA-EIA/105245/2008 and PTDC/EEAACR/69530/2006. HFD also thankfully acknowledges PhD fellowship from the same foundation, award SFRH/BD/45963/2008.

Author details
[1]Department of Bioinformatics and Computational Biology, The University of Texas M D Anderson Cancer Center, 1515 Holcombe Blvd Houston, TX 77030, USA. [2]Institute of Chemical and Biological Technology, Universidade Nova de Lisboa, Oeiras, Portugal. [3]Research Center for Intelligent Media, Furtwangen University, Furtwangen, Germany.

Authors' contributions
JSA identified the S3DB core model and drafted the manuscript. HFD developed the PHP prototype. WM uncovered and analyzed the model's logical patterns. All authors read and approved the final manuscript.

Received: 17 June 2010 Accepted: 20 July 2010 Published: 20 July 2010

References

1. Antezana E, Kuiper M, Mironov V: **Biological knowledge management: the emerging role of the Semantic Web technologies.** *Brief Bioinform* 2009, **10(4)**:392-407.
2. Chen H, *et al*: **Semantic web for integrated network analysis in biomedicine.** *Brief Bioinform* 2009, **10(2)**:177-92.
3. Cheung KH, *et al*: **Semantic Web for Health Care and Life Sciences: a review of the state of the art.** *Brief Bioinform* 2009, **10(2)**:111-3.
4. Crockford D: **JSONRequest, in JSON.org.** 2006.
5. Fong C: **How to access Web Services with GWT.** 2008 [http://www.gwtsite.com/how-to-access-web-services-with-gwt/].
6. Wikipedia: **JSONP.** 2010 [http://en.wikipedia.org/wiki/JSON#JSONP].
7. Richardson L, Ruby S: **RESTfull web services.** Sebastopol, CA, USA: O'Reilly Media IncLoukides M 2007.
8. Dahl R: **node.js.** 2009 [http://nodejs.org/].
9. Berners-Lee T: **Putting the Web back in Semantic Web.** 2005 [http://www.w3.org/2005/Talks/1110-iswc-tbl].
10. Backus J: **Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs.** *Communications of the ACM* 1978, **21(8)**:38.
11. Bell G, Hey T, Szalay A: **Computer science. Beyond the data deluge.** *Science* 2009, **323(5919)**:1297-8.
12. Hey T, Tansley S, Tolle K: **The Fourth Paradigm: Data-Intensive Scientific Discovery.** *Microsoft Research* 2009.
13. Neylon C: **Stitching science together.** *Nature* 2009, **461(7266)**:881.
14. Google. Google Wave Robots: **Overview.** *Google Wave API* 2009 [http://code.google.com/apis/wave/extensions/robots/].
15. Almeida JS, *et al*: **Data integration gets 'Sloppy'.** *Nat Biotechnol* 2006, **24(9)**:1070-1.
16. Davis JC, *et al*: **The microeconomics of personalized medicine: today's challenge and tomorrow's promise.** *Nat Rev Drug Discov* 2009, **8(4)**:279-86.
17. Deisboeck TS: **Personalizing medicine: a systems biology perspective.** *Mol Syst Biol* 2009, **5**:249.
18. Barker AD, *et al*: **I-SPY 2: an adaptive breast cancer trial design in the setting of neoadjuvant chemotherapy.** *Clin Pharmacol Ther* 2009, **86(1)**:97-100.
19. McPherson JD: **Next-generation gap.** *Nat Methods* 2009, **6(11 Suppl)**:S2-5.
20. Beckett D: **RDF/XML Syntax Specification.** *W3C Recommendation* 2004 [http://www.w3.org/RDF], 2009/08/29 v 1.189.
21. Brickley D, Guha RV: ***RDF Vocabulary Description Language 1.0: RDF Schema.*** *W3C Recommendation* 2004 [http://www.w3.org/TR/rdf-schema/].
22. Smith MK, Welty C, McGuinness DL: **OWL Web Ontology Language Guide.** *W3C Recommendation* 2004 [http://www.w3.org/TR/owl-guide/].
23. Deus HF, *et al*: **Adapting experimental ontologies for molecular epidemiology.** *AMIA Annu Symp Proc* 2007, 935.
24. Deus HF, *et al*: **A Semantic Web management model for integrative biomedical informatics.** *PLoS One* 2008, **3(8)**:e2946.
25. Freire P, *et al*: **Exploratory analysis of the copy number alterations in glioblastoma multiforme.** *PLoS One* 2008, **3(12)**:e4076.
26. Stanislaus R, *et al*: **RPPAML/RIMS: a metadata format and an information management system for reverse phase protein arrays.** *BMC Bioinformatics* 2008, **9**:555.
27. Wang X, Gorlitsky R, Almeida JS: **From XML to RDF: how semantic web technologies will change the design of 'omic' standards.** *Nat Biotechnol* 2005, **23(9)**:1099-103.
28. Berners-Lee T: **Semantic Web Road map.** *W3C* 1998 [http://www.w3.org/DesignIssues/Semantic.html].
29. Almeida JS: **Directing a web application, doc.s3db.org, to a s3db Deployment where additional Entities can be resolved.** 2009 [http://s3dbdoc.googlecode.com/hg/index.html?url=https://ibl.mdanderson.org/TCGA].

doi:10.1186/1471-2105-11-387
Cite this article as: Almeida *et al.*: S3DB core: a framework for RDF generation and management in bioinformatics infrastructures. *BMC Bioinformatics* 2010 **11**:387.

Chapter 3 | Case Studies for Biomedical Knowledge Integration

Chapter Outline

In the two publications that are included in this chapter, the core principles for biomedical data management that were identified and implemented in the S3DB model as necessary requirement for KOS in life sciences domains were applied in two integrative bioinformatics scenarios.

Section 1. H.F. Deus, D.F. Veiga, P.R. Freire, J.N. Weinstein, G.B. Mills, and J.S. Almeida, "**Exposing The Cancer Genome Atlas as a SPARQL endpoint**", *Journal of Biomedical Informatics*, vol. 43, Sep. 2010, pp. 998-1008.*

In the first scenario, an S3DB system was configured to expose data from The Cancer Genome Atlas (TCGA), a multi-institutional, multi-interdisciplinary project to study the molecular mechanisms affecting several types of cancer, as a SPARQL endpoint. The advantage in this approach to integrating data from TCGA was two-fold – on the one hand, the fine grained permission control supported by S3DB enabled complementary usages by those who could access the patient private data and those that could only access the molecular data elements; on the other hand, it was possible to effectively integrate the results from the analysis of TCGA glioblastoma multiforme with the diseasome dataset [88].

* The candidate developed the prototype application S3DB, the framework to automatically assign data elements from TCGA as entities of the S3DB Core Model, collected the data, assembled the integrative queries and wrote the manuscript.

Section 2. M.C. Correa, H.F. Deus, A.T. Vasconcelos, Y. Hayashi, J.A. Ajani, S.V. Patnana, and J.S. Almeida, "**AGUIA: autonomous graphical user interface assembly for clinical trials semantic data services**", *BMC medical informatics and decision making*, vol. 10, Oct. 2010. **

In the second scenario, S3DB was used to configure a highly dynamic domain-driven interface for gastro-intestinal clinical trial research. The distinction of the domain descriptors from its instantiation that is supported by the S3DB core model representation enabled the identification of the Gbox, the natural complement for the Tbox/Abox dichotomy, which specifies the set of design patterns that enable automatic assembly of interfaces, a critical requirement for widespread adoption of semantic web technologies by non-IT experts. It is the identification of different Gboxes that enables the multiple and often volatile views over the set of parameters that constitute a dataset.

** The candidate helped identify the Gbox, developed the prototype application S3DB that was used for data representation, helped devise the representation model and collect the data.

Journal of Biomedical Informatics 43 (2010) 998–1008

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Exposing the cancer genome atlas as a SPARQL endpoint

Helena F. Deus [a,b,*], Diogo F. Veiga [c], Pablo R. Freire [d], John N. Weinstein [a], Gordon B. Mills [c], Jonas S. Almeida [a]

[a] Department of Bioinformatics and Computational Biology, The University of Texas M. D. Anderson Cancer Center, 1515 Holcombe Blvd., Unit 1410, Houston, TX 77230-1402, USA
[b] Instituto de Tecnologia Química e Biológica, Universidade Nova de Lisboa, Av. da República, Estação Agronómica Nacional, 2780-157 Oeiras, Portugal
[c] Department of Systems Biology, The University of Texas M. D. Anderson Cancer Center, 7435 Fannin Street, Unit 950, Houston, TX 77030, USA
[d] Department of Molecular and Cell Biology, Baylor College of Medicine, One Baylor Plaza, Houston, TX 77030, USA

ARTICLE INFO

Article history:
Received 7 May 2010

Keywords:
TCGA
SPARQL
RDF
Linked Data
Data integration

ABSTRACT

The Cancer Genome Atlas (TCGA) is a multidisciplinary, multi-institutional effort to characterize several types of cancer. Datasets from biomedical domains such as TCGA present a particularly challenging task for those interested in dynamically aggregating its results because the data sources are typically both heterogeneous and distributed. The Linked Data best practices offer a solution to integrate and discover data with those characteristics, namely through exposure of data as Web services supporting SPARQL, the Resource Description Framework query language. Most SPARQL endpoints, however, cannot easily be queried by data experts. Furthermore, exposing experimental data as SPARQL endpoints remains a challenging task because, in most cases, data must first be converted to Resource Description Framework triples. In line with those requirements, we have developed an infrastructure to expose clinical, demographic and molecular data elements generated by TCGA as a SPARQL endpoint by assigning elements to entities of the Simple Sloppy Semantic Database (S3DB) management model. All components of the infrastructure are available as independent Representational State Transfer (REST) Web services to encourage reusability, and a simple interface was developed to automatically assemble SPARQL queries by navigating a representation of the TCGA domain. A key feature of the proposed solution that greatly facilitates assembly of SPARQL queries is the distinction between the TCGA domain descriptors and data elements. Furthermore, the use of the S3DB management model as a mediator enables queries to both public and protected data without the need for prior submission to a single data source.

1. Introduction

The Cancer Genome Atlas (TCGA) is a multi-institutional, cross-discipline effort led by the National Cancer Institute to characterize and sequence 20 cancer types at the molecular level [1]. The results, such as the discovery of new oncogenic mutations, come with the promise of clinically relevant population stratification and have recently been widened to form a coordinated international network of similarly minded initiatives [2]. TCGA is also a valuable resource for those interested in hypothesis-driven translational research as the bulk of its data results from direct experimental evidence. The level of complexity and detail of TCGA presents both an opportunity to statistically integrate the data [3] and a challenge in its representation. Heterogeneity and distribution of data sources are characteristics almost ubiquitous in biomedical datasets, which are often made available as data services without consistent data retrieval mechanisms and formats [4]. As such, advances in translational research often require complex infrastructures to integrate data from various autonomous sources and transverse several scientific domains [5]. Even when biomedical data are exposed as Web services, these tend to reflect the heterogeneity of the data, creating a challenge for its analysis with automated tools [6]. The communities of those producing and consuming biomedical data sources have mostly agreed that wide adoption of Web services that share common protocols can greatly improve data reuse and integration without the need to locally store large quantities of data [7,8]. The Linked Data best practices [9] include a collection of standards for publishing and connecting structured data on the Web that have matured to the point of providing a practical solution for the life sciences [10], namely through use of Resource Description Framework (RDF) as a data representation formalism and SPARQL as its query language [11–13].

* Corresponding author at: Department of Bioinformatics and Computational Biology, The University of Texas M. D. Anderson Cancer Center, 1515 Holcombe Blvd., Unit 1410, Houston, TX 77230-1402, USA.
E-mail addresses: mhdeus@mdanderson.org (H.F. Deus), dveiga@mdanderson.org (D.F. Veiga), freire@bcm.edu (P.R. Freire), jweinste@mdanderson.org (J.N. Weinstein), gmills@mdanderson.org (G.B. Mills), jalmeida@mdanderson.org (J.S. Almeida).

1532-0464/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2010.09.004

1.1. Resource Description Framework

RDF is a generic model that relies on two key assertions: (a) that everything is a resource referenced by a Universal Resource Identifier (URI) and (b) that every resource is part of a triple [11]. A key feature of RDF is the separation between content and presentation, which makes it useful for transversing a variety of domains, organizations and data structures [14,15]. Datasets may be converted to RDF by identifying their data elements, which are assigned to URIs, and formalizing their relationships as triples of URIs. Common vocabularies and terminologies, such as those made available by the National Center for Biomedical Ontology [16], are often used to link different datasets. Projects such as Dbpedia [17], Bio2RDF [18], Neurocommons [19], Diseasome [20,21] and others already provide a large amount of linked biomedical data available as RDF [22].

1.2. Sparql

SPARQL, the schema-free RDF query language, was designed to allow queries to be expressed across diverse data sources based on data properties and the relationships established with other data elements rather than on the physical location of the data [23]. SPARQL queries are constructs of one or more three-element graph patterns, such as "*?Person :hasName ?Name .*", each including a subject as the first element (*?Person*), a predicate as the second element (*:hasName*) and an object as the third element (*?Name*). SPARQL graph patterns support both variable elements (for example *?Person* and *?Name*) and non-variable elements (*:hasName*), where the prefix ":" indicates the Universal Resource Locator (URL) portion of a URI. The elements specific to the domain of discourse are typically the predicates (*:hasName*), which provide an anchor for the query. The solution to a SPARQL query is a directed labeled graph reusable in future queries. These properties make SPARQL endpoints, particularly those available as Web services, a very attractive solution for biomedical data services [22] given their recurrent need for data integration methodologies and shared queries [24]. Experimental biomedical data exposed as SPARQL endpoints can greatly facilitate discovery in the life sciences as each data source can be re-used as part of query federation approaches [25].

However, several problems have been identified that hamper exposure and query of data through SPARQL endpoints without extensive technical knowledge of RDF. Notably, SPARQL is a schema-free protocol; as such formulating a query usually requires some level of eye-parsing of the data, which hinders automation [26]. Tools such as MashQL [26] or Exhibit [27] have been developed to aid in the assembly of SPARQL queries by using the underlying RDF dataset structure.

1.3. Services for an integrative infrastructure

In this report we describe an infrastructure to expose the experimental data collected by the TCGA initiative as a programmatically accessible SPARQL endpoint. TCGA experimental datasets were broken into their fundamental data elements and assigned to entities of the Simple Sloppy Semantic Database (S3DB) management model [28]. S3DB defines entities and relationships using an RDF schema (RDFS) core model that enables encapsulation of RDF triples as part of a domain description, also represented as RDF triples [29]. This solution allows both the data elements and the description of the domain to have a representation in RDF, thereby supporting SPARQL queries formulated using the domain descriptors while targeting the data elements. It is worth noting that the processing of queries in the infrastructure developed overcomes the problems associated with a static RDF representation of the data by serializing SPARQL to S3DB's protocol and query language (S3QL). A graphical tool was developed that automatically assembles SPARQL queries while navigating the description of the domain and probing the properties of its instantiation. The intended end users of the system are researchers interested in biomarker discovery that require access to both molecular raw data and clinical covariates or researchers interested in linking their own datasets to TCGA. Usage is illustrated with a case study in which biomarker identification and its biological annotation are integrated with the Diseasome dataset [20,21]. The various components of the infrastructure are made available as Representational State Transfer (REST) Web services such that each component may be re-used independently.

2. Materials and methods

The Cancer Genome Atlas is a cancer genome characterization and sequencing project generating high-throughput molecular biology data about clinical samples. That data needs to be organized, integrated and analyzed in order to identify and characterize the genomic changes in 20 cancer types. A total of 500 samples from each type of tumor were collected, along with clinical and demographic covariates. Experiments were performed by 11 distinct genomic and sequencing characterization centers (GSCCs) to obtain data regarding miRNA expression, single nucleotide polymorphisms, exon expression, DNA methylation, copy number, trace-gene-sample relationships and somatic mutations. The publicly available TCGA datasets are deposited by individual genomic characterization and sequencing centers into a shared File Transfer Protocol (FTP) location (ftp1.nci.nih.gov).

2.1. S3DB core model URIs

The S3DB engine (http://s3db.org) was used to reassemble data elements from the TCGA initiative as RDF triples. The organizational model of S3DB defines a total of seven entities that define relationships between data elements. These are: Deployment, an entity representing an instance of an S3DB engine; User, an authenticated entity of any S3DB Deployment; Project, an entity that represents a specific domain by aggregating its entities and attributes; Collection, any entity associated with a domain that may be instantiated; Rule, the association between two Collections or between a Collection and a literal attribute; Item, an instance of a Collection; Statement, the relationship between two Items or between an Item and a literal value (see Fig. 4 in [28]). By design, each instance of an S3DB entity is automatically associated with a URI that consists of a URL (identifying the S3DB deployment in which the data are kept) concatenated with an alphanumeric identifier composed of the first character of the entity name (D, U, P, C, I, R or S) and a numeric component unique for each deployment of S3DB. The TCGA domain descriptors and their relationships, i.e. the metadata describing the data, were assigned to S3DB Collections and Rules, whereas the TCGA data elements and their attributes were assigned to S3DB Items and Statements. All assignment steps were performed using the S3DB protocol (S3QL), which supports select, insert, update and delete operations.

2.2. TCGA data structures

The TCGA datasets are made available through the TCGA portal (http://cancergenome.nih.gov/) as compact assemblies of data elements with various degrees of structure: as FTP directory structures, as eXtended Markup Language (XML) and as Microarray and Gene Expression Tabular (MAGE-tab) format. MAGE-tab is a spreadsheet-based, standard format for microarray data that

includes an investigation description format (IDF) file, which contains details about each experiment; a sample to data relationship format (SDRF) file, which describes the association of each sample with raw and processed data files and several files (.CEL or .txt) containing the experimental or analytical results [30]. Each format was handled separately during the process of assignment of TCGA data elements to S3DB entities. The data primer document [1] released by the TCGA consortia was used to assist in the interpretation of each archive name and code.

2.2.1. Formal representation of the TCGA workflow

The 3 TCGA dataset formats described are generated during the course of an experimental workflow to produce genomic characterization files (Fig. 1). The workflow consists of obtaining a genomic characterization element, corresponding to the latest revision of a raw data file, from a Sample, which in turn was collected from a Patient. Data elements involved in this workflow (Genomic Characterization, Sample and Patient) were assigned to S3DB Collections whereas the relationships established between them were assigned to S3DB Rules.

2.2.2. Semantic caching

Raw TCGA genomic characterization files are compacted and distributed as compressed archives through the TCGA initiative FTP server at ftp1.nci.nih.gov. It is not uncommon for a specific revision of a file to be requested more than once in order to replicate an analysis. To avoid the need for creating a local copy and reprocessing the same large TCGA archives each time a revision is required, a caching Web service was developed as part of the infrastructure described here so that each compressed archive is downloaded only once, uncompressed and stored locally. This procedure, illustrated in Fig. 2, dynamically iterates through the FTP directory structure to discover the appropriate file associated with a specific sample given a platform, an institution and a cancer type. Each raw data file was assigned to an S3DB Statement as a symbolic link (including the required attributes) in the form http://tcga.s3db.org/TCGA-sync.php?institution=[institution_url]&platform=[platform_code]&sample_id=[sample_id] &cancer_type=[cancer_type].

2.2.3. Mapping between the TCGA datasets and S3DB entities

Attributes associated with each TCGA data file are obtained by recursively navigating the FTP directory structure. The symbolic directory paths that terminate in files containing data are used to retrieve attribute-values for data elements concerning the genomic characterization center, array platform, data type and archive serial index (Fig. 3.1), which are assigned to values of S3DB Statements. For example, the symbolic directory path /tcga/tumor/gbm/cgcc/broad.mit.edu/ht_hg-u133a/transcriptome/ describes the content as originating from the "tcga" initiative, specifically from a "tumor" study in which the cancer type was glioblastoma multiforme ("gbm"), the sample was collected at the Broad Institute ("broad.mit.edu") Cancer Genomic Characterization Center ("cgcc") and the analytical platform Affymetrix HT Human Genome U133 Array Plate Set ("ht_hg-u133a") was used to generate "transcriptome" data.

Data generated by each of the participant genomic characterization centers for a given batch of patient samples and a given analytic platform are described in the MAGE-tab SDRF files, where a detailed listing of all the data files within an FTP archive can be found along with the associated sample barcodes. This index was used to establish a relationship between each raw data file and the corresponding sample (Fig. 3.2). The sample barcodes were

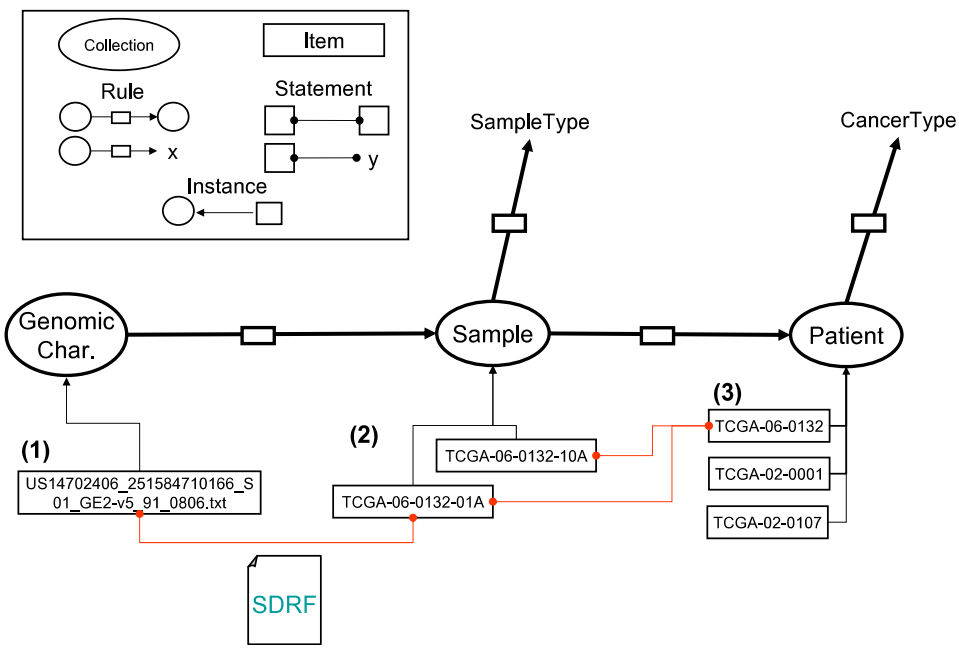


Fig. 1. Mapping the TCGA experimental workflow to S3DB entities. A Genomic Characterization element links a raw array data file containing either copy number or expression to a patient's clinical information (1–3). The filename syntax "US14702406_251584710166_S01_GE2-v5_91_0806.txt" (1) was used to link the raw data to the patient indirectly using the information in the SDRF file. In the example, the raw data is obtained from Sample "TCGA-06-0132-01A" (2), which was collected from a tumor (as indicated by "01A") of Patient "TCGA-06-0132" (3). Each of these links was assigned to an S3DB Statement whereas the links between domain descriptors "GenomicCharacterization", "Sample" and "Patient" were assigned to S3DB Rules.

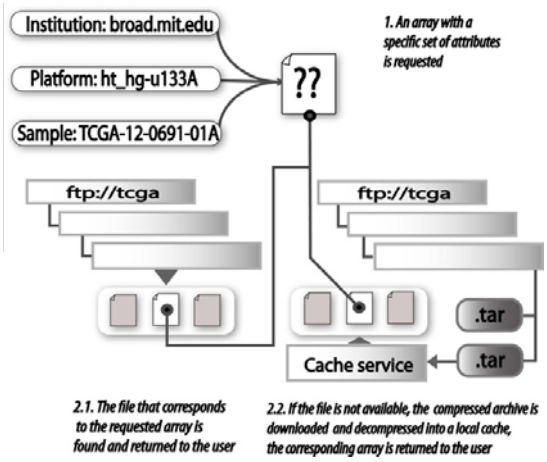


Fig. 2. A caching service for TCGA archives overcomes the need for bulk downloads. The caching service finds and retrieves the latest revision of a raw data file from the TCGA archives given an institution, a platform and a sample (1); the caching service will also retrieve a specific file revision if requested. The dynamic link generator, available at http://tcga.s3db.org/TCGAsync.php, recursively browses the TCGA datasets to return the raw data file corresponding to the requested array (2.1). If the archive has been compressed and the raw data file is not available as a symbolic link (2.2), then the TCGA archive is downloaded to the caching server, the archive is decompressed and the data file is returned. If the file has been requested previously, it is retrieved from the caching service.

separated into their constituent parts and used to characterize patient and sample metadata; as an example, the identifier of a sample is a 16-character barcode such as "TCGA-06-0132-01A," where the second alphanumeric portion, "06," was used to assign the appropriate sample collection center ("06" corresponds to the Henry Ford Hospital) and the last portion of the sample identifier, "01A," was used to assign the tumor type of the sample ("01" corresponds to a solid tumor, "A" indicates the first vial).

Patient clinical data for TCGA samples were retrieved from the XML data files, when available, to assign clinical parameters, including biospecimen collection center barcodes, tumor tissue site, sample primary metastatic status, histological type and tumor sample anatomic location.

2.3. Availability and documentation

The complete set of S3DB Rules describing the TCGA domain is available at http://ibl.mdanderson.org/TCGA/S3QL/rule/project_id=126 and graphically at http://tcga.s3db.org/map. TCGA data elements may be browsed through the S3DB graphical interface at http://tcga.s3db.org/login by using "public" as both the username and password.

The element assignment procedure described here (by recursively browsing the TCGA archives and extracting information from both the FTP directory structure, the SDRF files and the clinical data XML files) was developed in PHP and is available for download, along with documentation at http://code.google.com/p/tcga2s3db/.

PHP and an S3DB deployment are necessary to execute the application. S3DB may be downloaded from http://s3db.org, having as dependencies a Web server, mySQL and PHP. Documentation for the S3QL protocol used to assign data elements to the S3DB core model is available at http://s3db.org/documentation/s3qlsyntax.

All infrastructure components are available as REST-compliant Web services: the SPARQL endpoint is available at http://tcga.s3db.org/sparql.php and the semantic caching utility is available at http://tcga.s3db.org/TCGAsync.php. A graphic user interface for SPARQL assembly is available at http://tcga.s3db.org and the resulting RDFS document is available at http://tcga.s3db.org/rdf. Two third party tools were also configured for visualizing the TCGA RDF representation: an Exhibit [31] representation is available at http://tcga.s3db.org/exhibit and Allegrograph compatibility is demonstrated in the screencast at http://www.youtube.com/watch?v=BI5bf-taGU4.

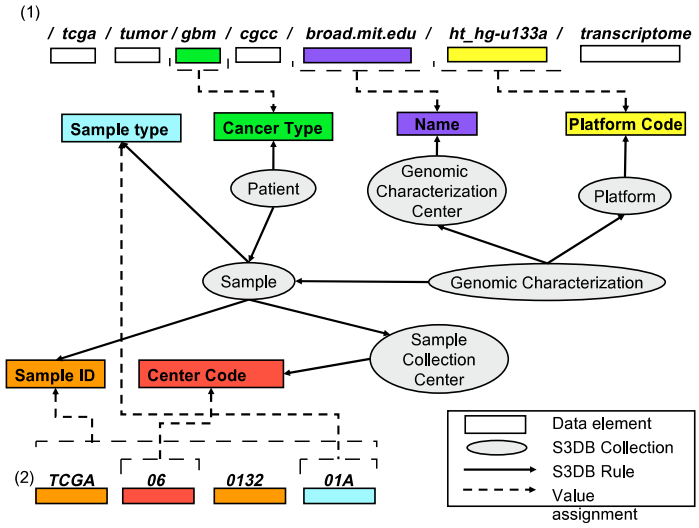


Fig. 3. Breaking TCGA data structures and assigning the data elements to S3DB statements. The path to raw data files (1) is separated into its constituent slash-separated portions, and the resulting data elements are assigned to values of S3DB Statements according to their position in the path. For example, the string "ht_hg-u133a" is assigned as the value for the "Platform Code" of an S3DB Statement concerning a specific Item of the Collection "Platform". Similarly, each sample barcode (2) is broken down into its constituent elements to retrieve values for Sample ID, Sample Type and Sample Collection Center Code, among others (not shown).

3. Results

3.1. Semantic Web services

The S3DB engine was used to mediate the exposure of experimental data from TCGA to Web services by assigning granular TCGA data elements to S3DB entities. As a consequence of this approach, two methodologies were made available that mediate the exposure of TCGA as a SPARQL endpoint: direct use of the exported data as an RDF document and serialization of SPARQL into S3DB's native query language and protocol (S3QL). Using the first method, all data that are part of an S3DB project are exported as an RDF document and queried using a SPARQL service, such as http://www.sparql.org/sparql.html with the URL for the RDF exported document (available at http://tcga.s3db.org/TCGA.rdf) in the "FROM" clause. However, Web-based repositories of experimental data, such as the TCGA datasets, are typically subject to updates, both in the amount of data and in their representation. The second method, in which the exposure of TCGA experimental data as a SPARQL endpoint is mediated by serializing SPARQL queries to S3QL, offers a flexible solution whereby data always reflect the latest state. For example, S3DB Statements that have been assigned to S3DB Rule *R44596*, which correspond to the "Name" attribute of a TCGA Sample Collection Center, are retrieved by the S3QL query http://ibl.mdanderson.org/TCGA/S3QL/statement/rule_id=44596. The resulting S3DB Statements are, essentially, triples in the form [*Item-Rule-Item*] or [*Item-Rule-Literal*]. Consequently, the S3QL query represented above has a SPARQL equivalent: *?SampleCollectionCenter :R44596 ?Name* ., which is formulated directly from the description of the S3DB Rule *R44596* used in the predicate : [*SampleCollectionCenter hasName Name* .]. Note that S3DB Rules are typically defined by the researchers producing the data through S3DB-associated interfaces and are not necessarily knowledgeable about Semantic Web technologies [29,32].

Fig. 4 illustrates a SPARQL query in which all genomic characterization arrays and corresponding Samples from patients treated at "MD Anderson" are retrieved. The query is assembled by mapping S3DB rules from the TCGA representation to SPARQL graph patterns and is executed through the SPARQL serialization Web service at http://tcga.s3db.org/sparql.php. The SPARQL serialization engine optimizes the time to a result by parallelizing and executing S3QL queries in stages, according to the amount of data that is expectable. For example, SPARQL patterns in which two of the three elements (subject, predicate and object) are constant, such as *?SampleCollectionCenter :R44596 'Henry Ford Hospital'*, are executed first as they return a small number of results. Furthermore, both the computed SPARQL query result and each serialized S3QL result are cached in order to improve query performance. This cache may be deleted by indicating "&clean=1" in the URL. More SPARQL queries on the TCGA domain are available at http://tcga.s3db.org and http://s3db.org/documentation/sparql.

Whenever applicable, URIs created by assignment of TCGA domain descriptors and data elements to S3DB entities were mapped to terms from widely used controlled terminologies such as MGED Ontology [33], OBI [34] and NCI thesaurus [35]; Bioportal [36] was used to discover the appropriate terminology equivalents and a specialized extension of the S3QL protocol was devised to support the mapping of TCGA URIs to controlled terminologies (see http://s3db.org/documentation/s3qlsyntax/#TOC-Dictionary).

3.2. SPARQL endpoint interface

An interface to support the use of the TCGA SPARQL endpoint was developed (Fig. 5, http://tcga.s3db.org/). It relies on the navigation of the TCGA domain rule set to facilitate the construction of SPARQL queries. The interface is populated directly from S3DB Rules, therefore changes in the description of the domain are immediately reflected in the interface. The action of selecting a Collection will display the attributes available for query in the "Rules" box. A "FROM" or "GRAPH" clause may be added to the SPARQL query, as described by [23], to integrate the results with data sources external to S3DB.

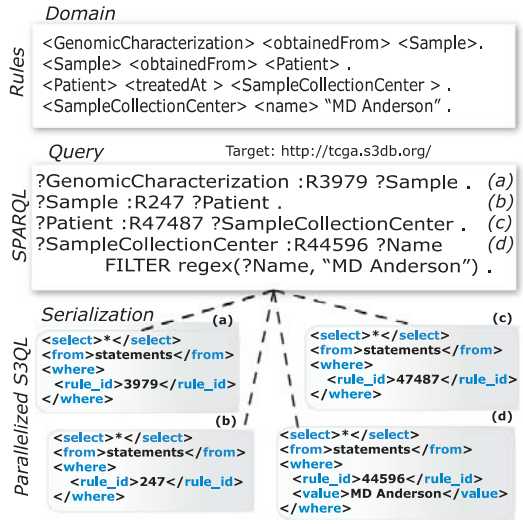


Fig. 4. Serialization and parallelization of a SPARQL query. The description of the domain as S3DB Rules is mapped into a SPARQL query by replacing the predicate of each rule with its identifier. Each graph pattern is then serialized to its equivalent S3QL query; for example, "*?GenomicCharacterization :R3979 ?Sample*" is translated into http://ibl.mdanderson.org/TCGA/S3QL.php?query=<S3QL><select>*</select><from>statement</from><where><rule_id>3979</rule_id></where></S3QL>&format=rdf, equivalent to http://ibl.mdanderson.org/TCGA/S3QL/statement/rule_id=3979, and executed in parallel, with the results intersected to produce a solution.

3.3. A copy number analysis use case

The advantage of exposing TCGA data to a SPARQL endpoint can be illustrated with the exploration of DNA copy number variation (CNV) in glioblastoma multiforme. The work presented in [3] presents a stand-alone tool that traverses, represents and analyzes CNV data for multiple tumor samples in real time using the TCGA data representation described in this report. The collection of data to perform such analysis in real time requires quick transversal of the TCGA datasets, an operation that would be challenging to integrate and correlate with clinical variables if data were retrieved directly from the TCGA FTP server. Alternatively, the data could be collected by performing a single SPARQL query in which the required parameters were configured, namely by setting the data type as "Copy Number results", the genomic characterization center as "mskcc" and the cancer type as "gbm". This query is illustrated in demo query Q2 at http://tcga.s3db.org. Performing the CNV analysis generated a list of 146 genes in aberrant regions, which were assigned to S3DB Statements in a related S3DB project termed *TCGA analytics* linked to the TCGA Samples described above (Fig. 3.2). This project includes as the main element an instance of *Analysis*, described as *Exploratory Analysis of the Copy Number Alterations in Glioblastoma Multiforme*, which functions as the intermediate between two lists: TCGA Samples that were analyzed and Genes that were found in aberrant regions. The *TCGA analytics* project can be visually explored at http://tcga.s3db.org/login.

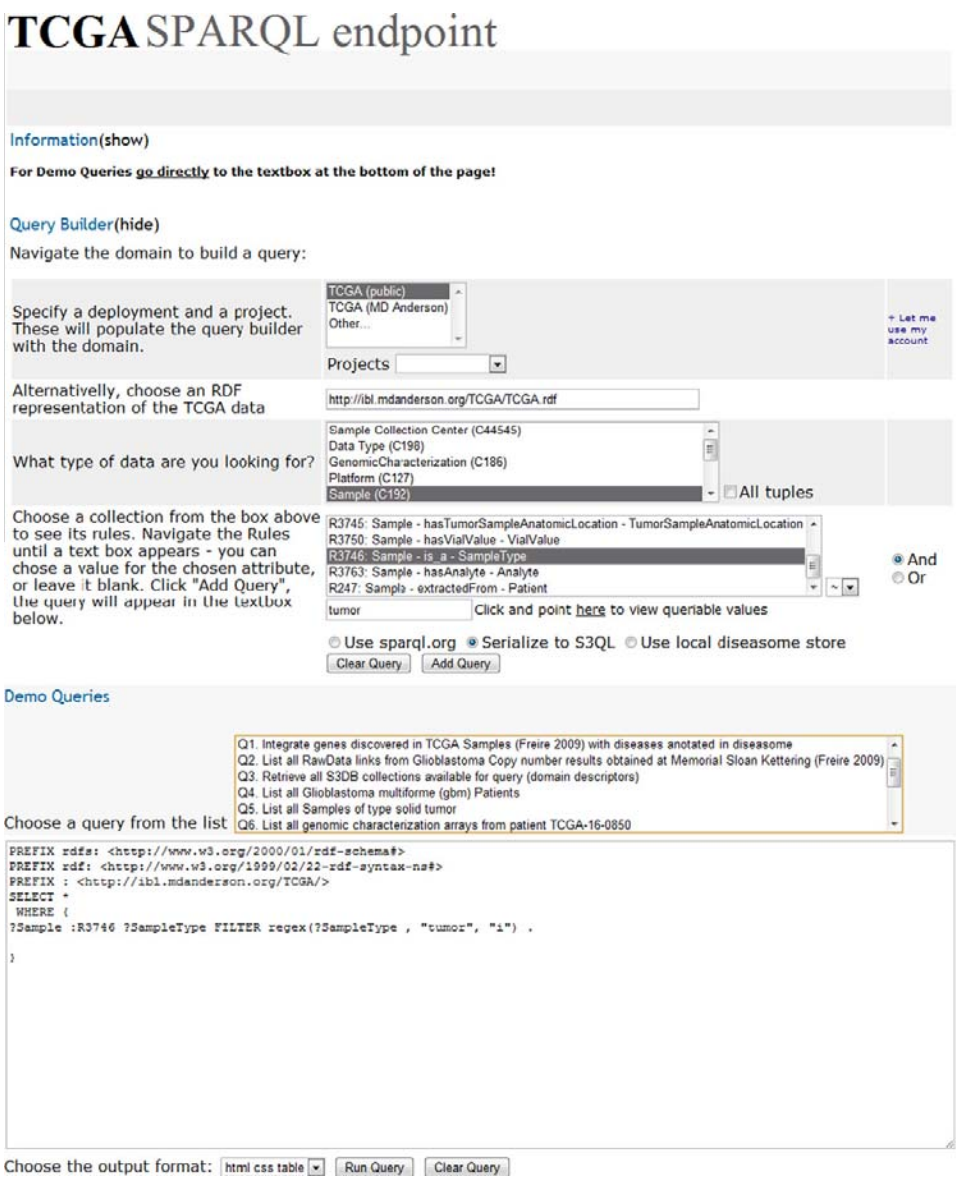


Fig. 5. A snapshot of the SPARQL interface for TCGA that automatically writes SPARQL queries by navigating the domain. Once an S3DB Collection is chosen, the Rules available for query will be displayed. Whenever an S3DB rule in which the object corresponds to an S3DB collection is chosen (for example, [Sample extractedFrom Patient]), a new rule menu is displayed. When the object of the rule is literal (for example, [Sample is_a SampleType]) a text box will appear in which a value for the chosen attribute may be fully matched (using "=") or partially matched (using "~"). The TCGA SPARQL endpoint is available at http://tcga.s3db.org.

Using the retrieved gene list, a SPARQL query was devised to discover associated diseases using data from the Diseasome dataset [20] (Fig. 6). Because the SPARQL endpoint service from the Diseasome project was found to be down with some frequency, for logistic reasons we have also downloaded the Diseasome N-triple statements into an RDF store using the ARC library for PHP. The query illustrated in Fig. 6 combines the Diseasome and the TCGA datasets by making use of the 'SERVICE' tag [37] to retrieve a source RDF graph from a remote SPARQL endpoint. This avoids the need to locally store each data source to be queried thereby enabling SPARQL federation without forcing a static RDF representation of the data. Data from the two datasets is linked by means of the National Center for Biotechnology Information (NCBI) gene symbol. From the resulting integrated data it can be observed that a total of 72 diseases are associated with the same genes discovered in aberrant regions of glioblastoma multiforme, the most common being leukemia and melanoma with 3 concurrent observations.

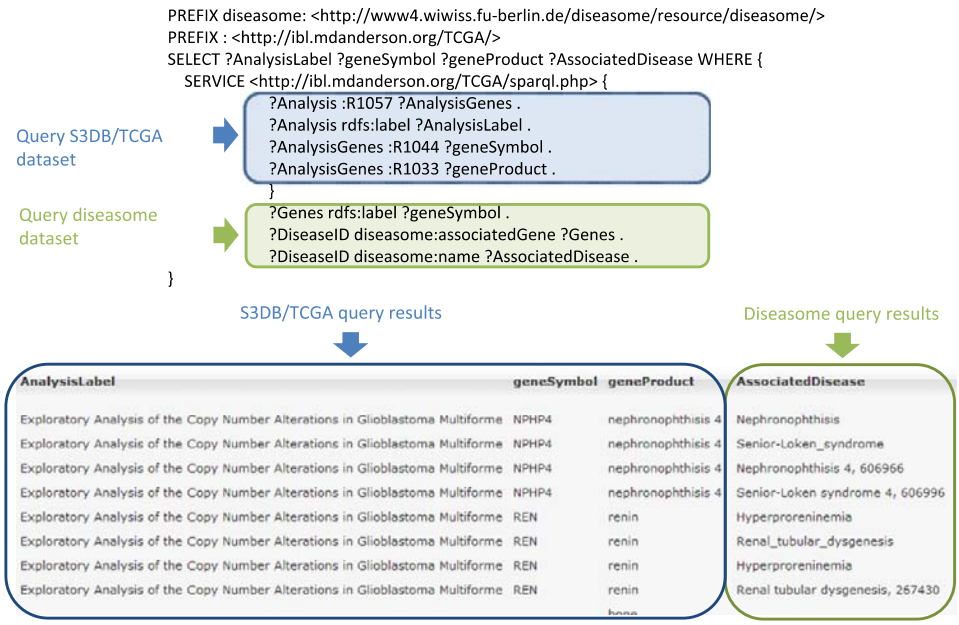


Fig. 6. A query to integrate TCGA derived gene list and the Diseasome dataset. The SPARQL 'SERVICE' tag is used to retrieve data from the TCGA SPARQL endpoint in real time without the need to create a local representation of the complete RDF graph. Illustrating the effective aggregation of the two distinct data sources, the output of the query includes both data collected from the S3DB/TCGA source and data from the diseasome project. Complete query results are available at http://tcga.s3db.org (demo query Q1).

3.4. System architecture

The overall architecture of the infrastructure developed including input and output components is depicted in Fig. 7. TCGA domain descriptors were manually assigned to S3DB Rules using S3DB-associated interfaces, and individual data elements were programmatically retrieved from 3 types of TCGA data structures (FTP directory, XML files and MAGE-tab format). The caching service, described in Fig. 2, was successfully tested as a buffer for scalability problems caused by the need to transfer large archives and frequent modifications in the structure of the FTP directory. The analytical results of the copy number variation use case were linked to the original samples providing the data by assignment to S3DB entities. It is worth noting that the Diseasome dataset used to discover diseases associated with aberrant genes in glioblastoma multiforme, described in Fig. 6, is not a component of the S3DB/TCGA system. Integration with the Diseasome dataset is achieved by using the proposed W3C standard "SERVICE" tag in the SPARQL query.

Integrated data may be retrieved from the S3DB engine either by obtaining the complete TCGA/RDF representation or through a SPARQL endpoint in which the query is primarily serialized into its S3QL equivalent (see Fig. 4). Additionally, data may be retrieved by direct use of S3QL (see http://s3db.org/documentation/s3qlsyntax for documentation on S3QL).

4. Discussion

An infrastructure has been developed to programmatically expose clinical and molecular data generated by The Cancer Genome Atlas project to Linked Data Web best practices, in particular as a SPARQL endpoint. The proposed solution makes use of the S3DB management model by assigning TCGA data elements and domain descriptors to entities of the S3DB RDF Schema core model. Specifically, the TCGA data elements were assigned to S3DB Statements, which in turn instantiate a separate set of domain descriptors, the S3DB Rules (Figs. 1 and 3). Resulting domain descriptors were mapped to terms from widely used controlled vocabularies and a Web application was developed to assemble REST-full SPARQL calls by navigating the description of the TCGA domain (Fig. 5).

The formal representation of experimental data from TCGA using the RDF model facilitates the task of data integration at various levels when compared to current data integration practices. One of the key benefits of the RDF model for data discovery is the reorganization of data according to its relevance in the domain rather than by content management needs. As an example, the most relevant data elements in TCGA, the raw data files that represent the outcome of a genomic characterization experiment, are retrieved by assembling a single intuitive SPARQL query such as the one in Fig. 4. Linked Data queries may thus be formulated in terms of the workflow pursued to collect the data, i.e. using SPARQL variables such as "?Sample" or "?Patient", rather than in terms of the infrastructure used to represent it. To retrieve those same data elements from the original portal, it would be necessary to browse through several data structures in several data processing steps. A second key benefit of using the RDF model is its flexibility in establishing links with datasets that were generated with different purposes. For example, integration of the TCGA experimental dataset and the Diseasome dataset collected by the Human Disease Network includes a list of 72 diseases that could potentially be related to glioblastoma multiforme because they share the same aberrant genes. A common impediment to the adoption of RDF in information management systems derives from the same decoupling of content and presentation that grant flexibility to RDF, which results in the absence of a clearly defined data schema to be used as an anchor for queries. Often some 'eye-parsing' of the data is required in order to formulate a query [26]. A key feature

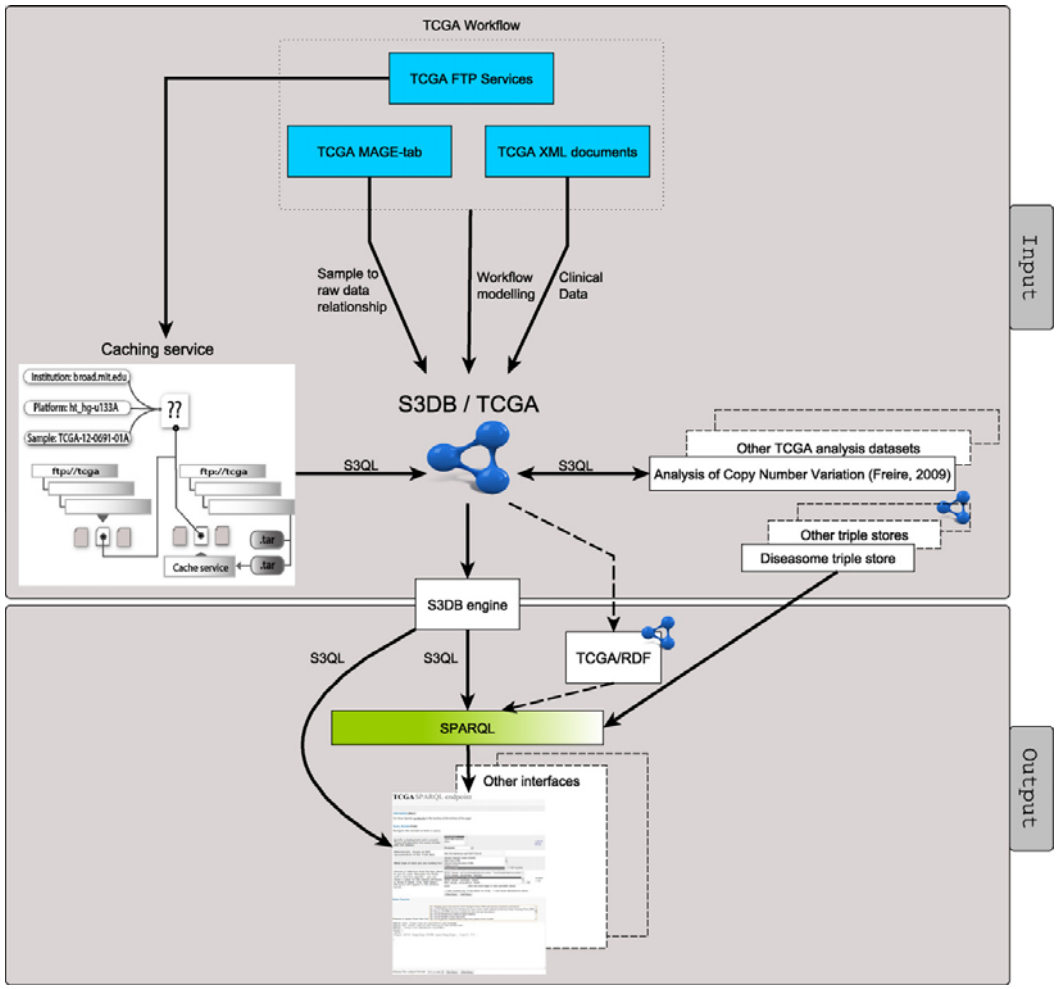


Fig. 7. Overview of the infrastructure developed to expose experimental data from The Cancer Genome Atlas as a SPARQL endpoint. The system components are divided into input and output components. The input components are directed mainly at breaking three types of TCGA data structures into their data elements while assigning them to S3DB entities. The output components are aimed at providing application programming interfaces for extracting data assigned to S3DB entities.

of the solution proposed here is the separation of the domain descriptors from their instantiation, provided by the assignment of data elements to the S3DB management model [28,29]. As an illustration, a graphical RDF representation tool, Sentient Knowledge Explorer, was used to generate Fig. 8 from the complete RDF graph of the TCGA datasets; the domain descriptors (yellow nodes) are clearly separate from the data elements (blue and grey nodes such as "Affymetrix HT Human Genome U133 Array Plate Set") because they are assigned to entities from the S3DB core model.

The S3DB constraint that requires S3DB statements (blue lines) to be instances of S3DB rules (black lines) is an intermediate step that greatly facilitates the relational algebra exercise of assembling SPARQL to retrieve TCGA contents. As an example, the SPARQL query presented in Fig. 4 has an intuitive syntax that is built from the description of the domain; that is, it emerges from mapping a user-defined Rule such as [Genomic Characterization obtained From Sample] (identified as R3979) to the SPARQL triple [*?GenomicCharacterization :R3979 ?Sample*]. The development of Web applications that generate SPARQL queries based on user-defined domains, an example of which is available at http://tcga.s3db.org/ (Fig. 5), therefore becomes an exercise of mapping the description of the domain, assigned to S3DB Rules, to SPARQL graph patterns.

An additional outcome of annotating TCGA domain descriptors to S3DB Rules is the creation of an intermediate layer between analytical applications and raw data. This prevents changes in the original TCGA FTP structure, such as compressing archives, from affecting data retrieval. An extreme example of rewiring an FTP directory would force only a small change in the automated assignment procedure; however, it would not affect query functionality. The description of S3DB Rules can also be freely edited, because the relationship between the domain descriptors and their data elements are established using alphanumeric identifiers rather than descriptive terms [29].

Finally, it is worth noting that by developing the Web service at http://tcga.s3db.org/ based on S3DB, an open-source biological management tool [38], it benefits not only from the REST protocols described but also from code portability, distribution with

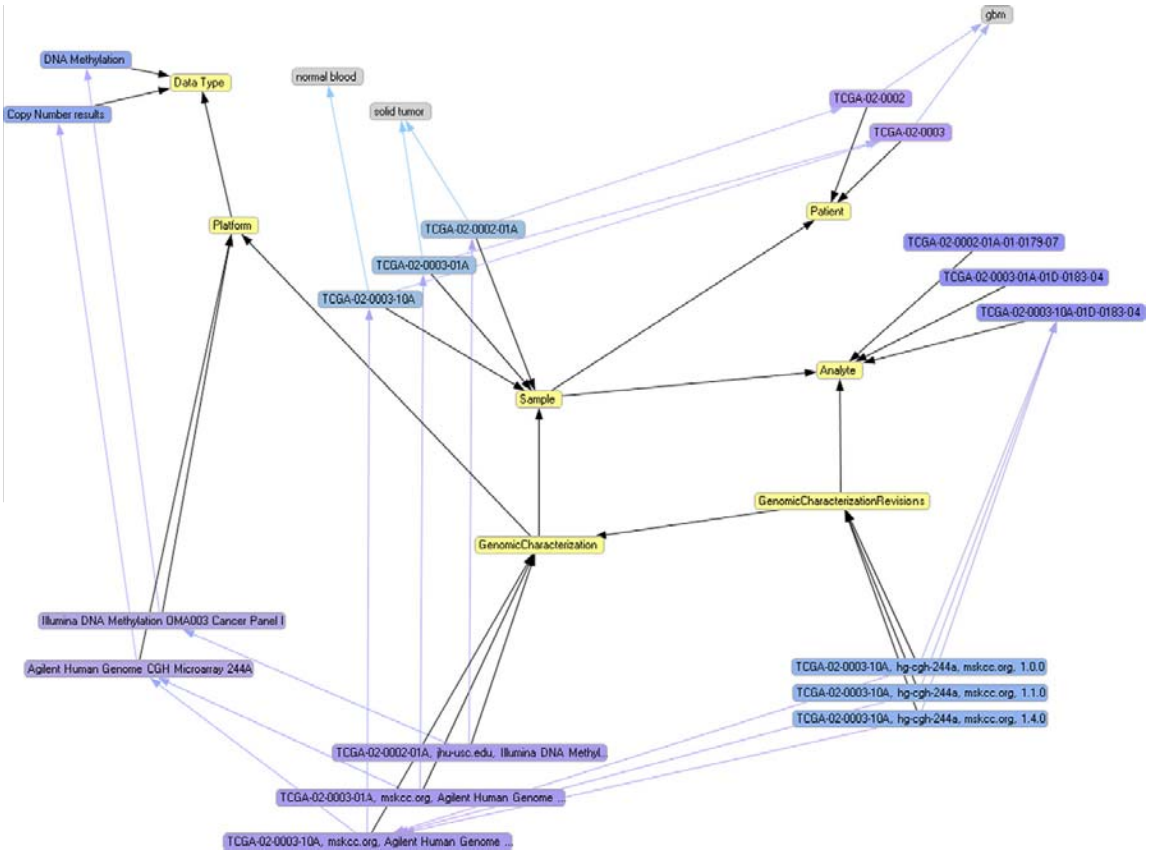


Fig. 8. A fragment of the RDF graph representation of TCGA created by the software tool Sentient Knowledge Explorer. Yellow nodes represent S3DB Collections, blue and purple nodes represent S3DB Items and grey nodes represent values of properties of the items. Light blue lines represent the relationships between items; black lines represent the connections between the elements of the core. Assignment of TCGA data elements to S3DB core elements results in a directed labeled graph in which the domain (yellow nodes) is clearly separate from its instantiation (blue and grey nodes). The separation of the domain facilitates the assembly of SPARQL queries, as it separates what is an actual data element (for example, sample "TCGA-02-0002-01A") from what is a domain descriptor (for example, the yellow node "Sample").

peer-to-peer interoperability and the availability of queries on protected data given the appropriate credentials (username and password), requirements that data management systems for biomedical domains must be able to juggle. Because the S3DB engine is distributed as open source, users of the system may choose to replace the default TCGA S3DB deployment with any S3DB deployment of their choice, where the domain may be freely configured and extended. Changes to S3DB Rules are immediately reflected in the SPARQL configuration tool, allowing researchers with a focus on TCGA dataset analysis to attempt alternative configurations that may be useful for their own purposes. Users of the system may opt to simultaneously query the public TCGA data and protected data that would otherwise not be exposed to the Linked Data Web. These features have made the S3DB system and its SPARQL endpoint an attractive data service solution for client-side analytical platforms such as the Cancer Genome Browser [3]. The availability of a SPARQL serialization engine enables data annotated to S3DB to be immediately available for query without the need for generating an RDF document. Researchers interested in integrating their own gene list results with the Diseasome dataset using the SPARQL service provided with every S3DB deployment need only assign their genes to S3DB statements.

4.1. System scalability and query performance

We have evaluated query performance of the system with a SPARQL query that was tested on TCGA data using both the SPARQL serialization engine described (see Section 3, Fig. 4) and a SPARQL engine without serialization using the complete TCGA/RDF representation as the data source. A screencast with these results is available at: http://www.youtube.com/watch?v=yrkA4uAT5GY. When the queries were executed simultaneously, the SPARQL serialization engine returned a result in approximately half the time required for the SPARQL engine without serialization.

4.2. Combining approaches to query federation

The cancer Biomedical Informatics Grid (caBIG®) is a project aimed at enabling the sharing of cancer-related data using a federated query model, whereby different institutions working on related problems can share data either by adapting their local repositories to a set of data models provided by caBIG or by adopting one of the caBIG applications [39]. The various data sources in caBIG can then be queried simultaneously using the caGrid query language [40]. The caBIG approach offers the advantage of facilitating query assembly because it relies on a set of common data

structures; this approach is therefore indicated for knowledge fields that are very well established. The linked data approach [9] does not impose a data model before integration is possible; instead data can be integrated using SPARQL queries as soon as an RDF representation is available. Although the latter approach requires that datasets be linked through the use of common terminologies before the assembly of SPARQL queries, it is better suited for knowledge areas that evolve quickly as it benefits from having novel data immediately available for integration. The two approaches could therefore greatly benefit from each other; indeed, a call for Semantic Web opportunities has been launched by the caBIG community [41]. The semCDI [42] and Corvus [43] projects, for example, have already developed extensive work towards modeling and integrating the various data models available at caBIG including the availability of SPARQL engines. The architecture described in this report could thus be easily integrated with caBIG datasets that are made available as RDF or if a SPARQL endpoint is provided. Indeed, one of the steps in that direction was mapping the terms within TCGA S3DB representation to NCI Thesaurus, also widely used by the caBIG community [44,45].

4.3. Limitations to the proposed solution

The work presented here attempts to provide a formal linked representation of the TCGA datasets that can be queried using SPARQL. However, datasets that are unavailable to the public may not be included in the RDF representation due to the academic nature of this work.

TCGA domain elements were mapped to biomedical ontologies whenever possible using RDF Schema and Web Ontology Language. We believe that this mapping is necessarily incomplete because it can only reflect a snapshot of currently available ontologies and will most likely not address all the needs of application development. As a consequence, the dictionary extension to the S3QL protocol was devised to enable extending the mapping beyond the S3DB schema, while still enabling mapped resources to be queried through the SPARQL serialization engine.

5. Author contributions

HFD and JSA designed the application; HFD developed the application, ran the challenges to the domain representation and wrote the report; DFV, PRF, JNW and GBM challenged the domain representation with examples and used the results to identify significant improvements to its topology.

Acknowledgments

We thankfully acknowledge IO-informatics for providing a Sentient Knowledge Explorer license to the Integrative Bioinformatics Laboratory of The University of Texas M. D. Anderson Cancer Center. We thank Rebecca Partida for reviewing the manuscript. We would also like to acknowledge two anonymous reviewers, who provided valuable insights towards the improvement of this report.

This work was funded by the Fundação para a Ciência e Tecnologia and the Center for Clinical and Translational Sciences under contracts SFRH/BD/45963/2008 and 1UL1RR024148, respectively. The work was also supported in part by the National Heart, Lung and Blood Institute and by the National Cancer Institute of the US National Institutes of Health under contracts N01-HV-28181 and P50 CA70907, respectively.

References

[1] TCGA Data Primer Version 1.0. <http://tcga-data.nci.nih.gov/docs/TCGA_Data_Primer.pdf>.
[2] The International Cancer Genome Consortium. International network of cancer genome projects. Nature 2010;464:993–8.
[3] Freire P, Vilela M, Deus H, Kim Y-W, Koul D, Colman H, et al. Exploratory analysis of the copy number alterations in glioblastoma multiforme. PLoS ONE 2008;3:e4076.
[4] Stephens SM, Rung J. Advances in systems biology: measurement, modeling and representation. Curr Opin Drug Discov Devel 2006;9:240–50.
[5] Ruttenberg A, Clark T, Bug W, Samwald M, Bodenreider O, Chen H, et al. Advancing translational research with the Semantic Web. BMC Bioinformatics 2007;8(Suppl. 3):S2.
[6] Vandervalk BP, McCarthy EL, Wilkinson MD. Moby and Moby 2: creatures of the deep (web). Brief Bioinform 2009;10:114–28.
[7] Baker CJO, Cheung K-H. Semantic web: revolutionizing knowledge discovery in the life sciences. New York: Springer; 2007.
[8] Goble C, Stevens R, Hull D, Wolstencroft K, Lopez R. Data curation + process curation=data integration + science. Brief Bioinform 2008;9:506–17.
[9] Linked Data. <http://www.w3.org/DesignIssues/LinkedData.html>.
[10] Stephens S, Morales A, Quinlan M. Applying Semantic Web technologies to drug safety determination. Ieee Intell Syst 2006;21:82–6.
[11] Semantic Web Roadmap. <http://www.w3.org/DesignIssues/Semantic.html>.
[12] W3C. Semantic Web Best Practices and Deployment Working Group. <http://www.w3.org/2001/sw/BestPractices/>.
[13] Wang X, Gorlitsky R, Almeida JS. From XML to RDF: how semantic web technologies will change the design of 'omic' standards. Nat Biotechnol 2005;23:1099–103.
[14] Rodriguez MA, Watkins JH, Bollen J, Gershenson C. Using RDF to model the structure and process of systems. InterJ Complex Sys 2007;2131.
[15] Powers S. Practical RDF: solving problems with the resource description framework. O'Reilly & Associates Inc.; 2003.
[16] Rubin DL, Lewis SE, Mungall CJ, Misra S, Westerfield M, Ashburner M, et al. National Center for Biomedical Ontology: advancing biomedicine through structured organization of scientific knowledge. Omics 2006;10:185–98.
[17] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: a nucleus for a web of open data. Semantic Web, Proc 2007;4825:722–35.
[18] Belleau F, Nolin MA, Tourigny N, Rigault P, Morissette J. Bio2RDF: towards a mashup to build bioinformatics knowledge systems. J Biomed Inform 2008;41:706–16.
[19] Ruttenberg A, Rees JA, Samwald M, Marshall MS. Life sciences on the Semantic Web: the neurocommons and beyond. Brief Bioinform 2009;10:193–204.
[20] D2R Server publishing the Diseasome Dataset. <http://www4.wiwiss.fu-berlin.de/diseasome/>.
[21] Goh KI, Cusick ME, Valle D, Childs B, Vidal M, Barabasi AL. The human disease network. Proc Natl Acad Sci USA 2007;104:8685–90.
[22] Goble C, Stevens R. State of the nation in data integration for bioinformatics. J Biomed Inform 2008;41:687–93.
[23] SPARQL Query Language for RDF – W3C Recommendation. <http://www.w3.org/TR/rdf-sparql-query/>.
[24] Cheung KH, Prud'hommeaux E, Wang Y, Stephens S. Semantic Web for Health Care and Life Sciences: a review of the state of the art. Brief Bioinform 2009;10:111–3.
[25] Cheung KH, Frost HR, Marshall MS, Prud'hommeaux E, Samwald M, Zhao J, et al. A journey to Semantic Web query federation in the life sciences. BMC Bioinformatics 2009;10(Suppl. 10):S10.
[26] Jarrar M, Dikaiakos MD. MashQL: a query-by-diagram topping SPARQL. In: Proceeding of the 2nd international workshop on ontologies and information systems for the semantic web. Napa Valley, California, USA: ACM; 2008.
[27] Exhibit: Publishing Framework for Data-Rich Interactive Web Pages. <http://www.simile-widgets.org/exhibit/>.
[28] Deus HF, Stanislaus R, Veiga DF, Behrens C, Wistuba II, Minna JD, et al. A semantic web management model for integrative biomedical informatics. PLoS ONE 2008;3:e2946.
[29] Almeida JS, Chen C, Gorlitsky R, Stanislaus R, Aires-de-Sousa M, Eleuterio P, et al. Data integration gets 'Sloppy'. Nat Biotechnol 2006;24:1070–1.
[30] Rayner TF, Rocca-Serra P, Spellman PT, Causton HC, Farne A, Holloway E, et al. A simple spreadsheet-based, MIAME-supportive format for microarray data: MAGE-TAB. BMC Bioinformatics 2006;7:489.
[31] Huynh DF, Karger DR, Miller RC. Exhibit: lightweight structured data publishing. In: Proceedings of the 16th international conference on World Wide Web. Banff, Alberta, Canada: ACM; 2007.
[32] S3DB documentation: webS3DB. <http://s3db.org/documentation/webs3db>.
[33] Stoeckert CJ, Parkinson H. The MGED ontology: a framework for describing functional genomics experiments. Comp Funct Genomics 2003;4:127–32.
[34] The ontology for biomedical investigations. <http://obi-ontology.org/page/Main_Page>.
[35] Sioutos N, de Coronado S, Haber MW, Hartel FW, Shaiu WL, Wright LW. NCI thesaurus: a semantic model integrating cancer-related clinical and molecular information. J Biomed Inform 2007;40:30–43.
[36] Noy NF, Shah NH, Whetzel PL, Dai B, Dorf M, Griffith N, et al. BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Res 2009;37:W170–3.
[37] SPARQL 1.1 Federation Extensions. <http://www.w3.org/2009/sparql/docs/fed/service#introduction>.
[38] Antezana E, Kuiper M, Mironov V. Biological knowledge management: the emerging role of the Semantic Web technologies. Brief Bioinform 2009;10:392–407.

[39] caBIG-Community. An Introduction to caGrid Technologies and Data Sharing In.
[40] caGrid Query Language (CQL) documentation. <http://cagrid.org/display/dataservices/CQL>.
[41] caBIG Semantic Web Opportunities. <https://wiki.nci.nih.gov/display/VCDE/caBIG+Semantic+Web+Opportunities>.
[42] Shironoshita EP, Jean-Mary YR, Bradley RM, Kabuka MR. SemCDI: a query formulation for semantic data integration in caBIG. J Am Med Inform Assoc 2008;15:559–68.
[43] McCusker JP, Phillips JA, Gonzalez Beltran A, Finkelstein A, Krauthammer M. Semantic web data warehousing for caGrid. BMC Bioinformatics 2009;10(Suppl. 10):S2.
[44] Cimino JJ, Hayamizu TF, Bodenreider O, Davis B, Stafford GA, Ringwald M. The caBIG terminology review process. J Biomed Inform 2009;42:571–80.
[45] caBIG Enterprise Vocabulary Services (EVS). <https://cabig.nci.nih.gov/concepts/EVS/>.

Correa et al. BMC Medical Informatics and Decision Making 2010, 10:65
http://www.biomedcentral.com/1472-6947/10/65

BMC Medical Informatics & Decision Making

TECHNICAL ADVANCE Open Access

AGUIA: autonomous graphical user interface assembly for clinical trials semantic data services

Miria C Correa[1,2*], Helena F Deus[1,3], Ana T Vasconcelos[2,4], Yuki Hayashi[5], Jaffer A Ajani[5], Srikrishna V Patnana[6,7], Jonas S Almeida[1]

Abstract

Background: AGUIA is a front-end web application originally developed to manage clinical, demographic and biomolecular patient data collected during clinical trials at MD Anderson Cancer Center. The diversity of methods involved in patient screening and sample processing generates a variety of data types that require a resource-oriented architecture to capture the associations between the heterogeneous data elements. AGUIA uses a semantic web formalism, resource description framework (RDF), and a bottom-up design of knowledge bases that employ the S3DB tool as the starting point for the client's interface assembly.

Methods: The data web service, S3DB, meets the necessary requirements of generating the RDF and of explicitly distinguishing the description of the domain from its instantiation, while allowing for continuous editing of both. Furthermore, it uses an HTTP-REST protocol, has a SPARQL endpoint, and has open source availability in the public domain, which facilitates the development and dissemination of this application. However, S3DB alone does not address the issue of representing content in a form that makes sense for domain experts.

Results: We identified an autonomous set of descriptors, the GBox, that provides user and domain specifications for the graphical user interface. This was achieved by identifying a formalism that makes use of an RDF schema to enable the automatic assembly of graphical user interfaces in a meaningful manner while using only resources native to the client web browser (JavaScript interpreter, document object model). We defined a generalized RDF model such that changes in the graphic descriptors are automatically and immediately (locally) reflected into the configuration of the client's interface application.

Conclusions: The design patterns identified for the GBox benefit from and reflect the specific requirements of interacting with data generated by clinical trials, and they contain clues for a general purpose solution to the challenge of having interfaces automatically assembled for multiple and volatile views of a domain. By coding AGUIA in JavaScript, for which all browsers include a native interpreter, a solution was found that assembles interfaces that are meaningful to the particular user, and which are also ubiquitous and lightweight, allowing the computational load to be carried by the client's machine.

Background

The heterogeneity of data produced by biomedical research creates a serious challenge to the interoperability and consistent aggregation of data [1], which renders the development and maintainance of web applications correspondingly more time consuming and resource intensive [2]. This reinforces a preference for front-end applications that are automated and web-based as much as possible. The semantic web resource description framework (RDF) offers particular advantages in this regard, as its data structure can contain a combination of domain ontology components as well as the graphic rules ontology. The domain ontology predates the work described here in the sense that it was previously identified and is regularly and independently changed by the domain experts. The regular changes in the domain ontology typically reflect new data sources, but may also correspond to a novel understanding of an old relationship between data elements. We proposed such an

* Correspondence: mcoelho@mdanderson.org
[1]Department of Bioinformatics and Computational Biology, The University of Texas MD Anderson Cancer Center, 1515 Holcombe Blvd, Houston, TX 77030, USA
Full list of author information is available at the end of the article

ontology incubation process in 2006 [3] and a formal model in 2010 [4]; an example of its application to lung cancer research illustrates the use of the simple sloppy semantic database (S3DB) tool to identify and maintain the correspondingly fluid RDF stores [5]. By contrast, the AGUIA ontology described in this report is fixed: it is designed to mediate the automated presentation of the fluid domain ontology and is therefore independent from the domain ontology. The instantiation of the AGUIA ontology involves the specification of the links between the graphical user interface (GUI) ontology and the domain ontology. Where this association is not specified, AGUIA will use default values to produce a default and a rather dull interface that will respond to the topology of the domain ontology. The dependency between S3DB collections and items (tabs), and rules and statements (rows) for the domain ontology is graphically captured as tabbed navigation. Therefore, the context for the automated interface assembly work described in this report is that of accounting for user bias/context in the representation of a knowledge base. In other words, the domain ontology serves as an underlying structure to describe the data content of the web application that describes the concepts and their relationships in the specified domain. This ontology was created to store the data from clinical trials for gastrointestinal cancer. The graphic model is based on graphic rules that describe the graphic structure of the web application. This model contains the graphic components of the web application and their association with the domain ontology. The web application is able to assemble a GUI through this RDF data structure. The supplementary material contains more details about the structure of data entry and the construction/instantiation of each component of the model. It includes a video demonstrating how the same AGUIA application is alternatively pointed to different S3DB deployments (see Materials).

The web application AGUIA was developed to manage clinical data through clinical trials semantic data services. Semantic data services are data objects with domain-specific semantics and technology standards that are used to provide secure real-time access to existing data sources. In addition to providing the context for sharing information based on program needs, semantic data services also support the dispersed data ownership requirement that generally exists for these programs. In a nutshell, semantic data services allow for the discovery and management of semantic relationships across information systems in a timely manner and on a large scale [6]. In this context, the clinical trials semantic data services (CTSDS) developed within S3DB deployments are semantic data services (and include a SPARQL endpoint) directed toward clinical trials. Through the CTSDS, it is possible to capture common characteristics of a particular group of patients and generate results that can best be examined by the physician. For example, it is possible to capture the data for all patients who smoke and have the same tumor type, such as a grade 2, moderately differentiated tumor.

The goal of developing a specification and standard that autonomously assembles a graphical user interface for a data source has a tradition in web technologies that can be traced back as far as hypertext in the use of HTML elements to manipulate the web browser's document object model.

The emergence of semantic web technologies has not entirely overlooked the assembly of user interfaces. This is particularly clear in RDF elements such as rdfs:label or rdfs:comment [7]. These elements anticipate the need for interface descriptors by other semantic models. Upon close scrutiny of the specifications, one finds that such elements are indeed widely used. For example, in W3C's simple knowledge organization systems (SKOS), the definition of the relationship skos:broader [8] comes with the indication that its rdfs:label is "has broader." SKOS [9] is a simple RDF schema for knowledge organization systems (KOS) such as thesauri, classification schemes, subject heading systems and taxonomies within the framework of the semantic web. Like many other RDFS-Plus [10] modeling efforts, encoding this information in RDF is ultimately geared toward the facilitation of interoperability between computer applications by creating a data infrastructure ecosystem in which the semantics of data generation and usage are both explicit in the representation [11].

The initial motivation for the work described here was the development of a web application to manage clinical data within the Gastrointestinal Medical Oncology Department of MD Anderson Cancer Center. It was clear from the beginning of the project that the diversity of users/usages and the fluidity of the underlying data model were not compatible with a conventional web application with a static layout. This is because the underlying schema are not only continuously undergoing significant changes, but those changes are not always the same for all involved. Therefore, it was decided that the ideal interface would be an autonomous browser-based web application that would respond to one or more independent descriptions of the data schema and graphic model in order for it to be used in multiple projects/representations. To enable the automatic assembly of the web application, a set of RDF descriptors was created by re-sorting the knowledge model schema [4] of an S3DB database [3]. The web application would then be able to use these descriptors together with the ontology that is the clinical data project, which is also stored on S3DB, to assembly itself.

This rationale follows a proposal by other researchers that a knowledge base requires more than terminology and assertion components (TBox and ABox, respectively). Translated to correspond with a modern emphasis on semantic web formalisms, the RDF descriptors could also include a GBox component. Whereas ABox and TBox are terms coming from description logic [12], GBox was proposed by Motik [13] to describe how that information should be displayed.

In the work described here, instead of seeking to identify high-level graphic relationships to maximize reusability, we have taken the opposite route. First, a suite of graphical user interfaces was developed in response to specific requests from clinical researchers, and only then was an effort made to identify design patterns that could be captured as a specialized RDF protocol. The justification for this reverse approach is that starting from a graphic configuration that was already meaningful in relation to a specific domain will lead to a set of descriptors that are more easily recognized - and used - by those domain experts when they engage information management systems.

In spite of the domain-rooted approach followed at the beginning of this work, the clinical-trial-driven interfaces also represent a starting point for a more generic formalism-oriented approach that was originally proposed for intelligent information presentation systems (IIPS) [2]. That early work found a natural extension in the emergence of extended markup languages and subsequently in semantic web formalisms, leading to efforts to model web site interfaces such asWebML [14] and OntoWebber [15]. Furthermore, as noted by Lei *et al.* [2], those models could be enhanced by also considering key user interface issues such as page layouts and graphical user interfaces. AGUIA considers these interface issues, such that an interface generated by AGUIA should allow for multiple user-specific layouts.

Methods

Methods: browser-based application development

AGUIA was developed using dynamic HTML (DHTML) concepts and technologies to create dynamic and interactive web pages. DHTML is not a programming language, but rather is a set of programming techniques that combine HTML, JavaScript, HTML DOM and CSS. DHTML enables dynamic elements to be created inside the web page: fine-grained configurations such as text, page styles (font color, size and others), element positions, etc., can be changed dynamically after the page is loaded [16]. The application delivered with this report makes extensive use of the dhtmlgoodies library [17], which provides low level support for basic graphic features such as calendars, tabs, folder trees and others. The Google code management system [18] provides open source hosting of this application, which is publicly available at http://aguia.googlecode.com/hg/index.html. All code is made available in the corresponding parent directory structure, with the open source project management tools available at http://aguia.googlecode.com.

Testing and evaluation was performed by tracking usage and response times. Screencasts of typical usage were recorded and are provided with this report (see Results). Note that in the S3DB system (see Materials), data modeling is performed by the users themselves through the definition of S3DB rules.

Materials: semantic database web service

The S3DB, which we used as our database web service, is an infrastructure for distributed data servicing that relies on semantic web concepts for the bottom-up management of heterogeneous domains [4]. It provides a bridge between a mass of structured data annotated by using personal ontologies and a globally referenceable semantic representation indexed to controlled vocabularies [4]. The S3DB web service exposes its API through a read/write REST protocol, S3QL. representational state transfer (REST) is a coordinated set of architectural constraints that attempts to minimize latency and network communication, while at the same time maximizing the independence and scalability of component implementations. This is achieved by placing constraints on connector semantics, in contrast to other styles that focus on component semantics. REST enables the caching and reuse of interactions, dynamic substitutability of components, and processing of actions by intermediaries in order to meet the needs of an Internet-scale distributed hypermedia system [19]. The S3DB database is also capable of producing its output in a variety of formats such as tabular text, XML and JSON, in addition to the generation of RDF in n3 or XML [20]. The open source application is made publicly available for a variety of operating systems. The RDF language provides a simple and flexible way of representing knowledge by breaking data structures down into dyadic predicates (triples) [21]. In other words, RDF is a general-purpose language for representing information in the web [7] and providing interoperability between otherwise incompatible domain models and formats. The RDF has an official query language created by W3C, the SPARQL. The S3DB is also able to receive SPARQL queries, which are then internally converted into S3QL [22], as recently illustrated for the cancer genome data [23].

Ultimately, the S3DB web service was developed to test the hypothesis that a user-editable schema and streamlined interoperability would facilitate the acquisition of biomedical data within the biomedical context and by the biomedical domain experts themselves [3].

The S3DB deployment, and by association its query language, S3QL, uses SQL to operate a regular relational database backbone. Most existing S3DB deployments use postgreSQL or mySQL. Other SQL databases that have been tested do not appear to pose a major limitation at that level. The performance is that of the supporting database, to which we add an overhead of migrating user permissions between entities of the S3DB data model. An online tool, available at http://s3db-operator.googlecode.com, illustrates the inner workings of this last component.

The research prototype of this database is mature enough that a few deployments have been adopted by MD Anderson Cancer Center and are subject to the same strict security audits of any other research tool dealing with sensitive data: see "Internal Services" at http://bioinformatics.mdanderson.org/. Also a number of external, public tools use it to service large datasets such as those produced by the cancer genome atlas (TCGA). For an example, see the DNA copy number browser at http://cnviewer.googlecode.com/. Currently, the scalability of this web service is that of the relational backbone - which is indeed more stringent than, for example, a map-reduction store.

Additional, non-essential I/O, server-side functionality was achieved by developing small applications coded in the PHP language. An example is a function to export the application data to excel spreadsheets, as most browsers don't offer native support for ActiveX technology. As noted in the previous section, all essential components of AGUIA are native to the web browser - that is, they are coded entirely in JavaScript.

Results

RDF protocol and list of graphic rules

The automatic assembly of the interface relies on two data sources, one being the target observational data and the other containing the graphic rules for the assembly of the interface. The graphic rules contain two parts: GUI actions and GUI rules (Figure 1), both of which are collections of an S3DB project. S3DB collections are described according to the RDF triple - *subject-predicate-object* - and S3DB rules are the subject or object of the RDF triple [4]. The GUI actions collection contains the actions that can be created. These are described in Figure 2. The GUI rules collection contains the instantiation of each graphic component. This collection contains the rules domain, range, action, trigger and value.

- Domain will contain the ID of the S3DB rule that will command the action.
- Range will contain the ID of the rule that will receive the action.

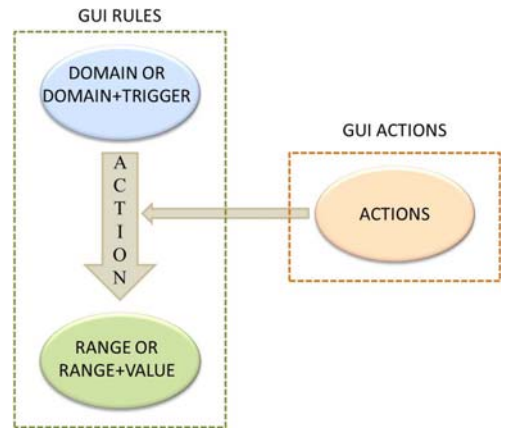


Figure 1 Relationship between GUI rules and GUI actions. Illustration of the relationship between GUI rules and GUI actions to generate RDF triples: GUI rules are S3DB rules predicated by S3DB items of the S3DB collection of AGUIA actions.

- Action will contain the action inserted in the GUI actions collection.
- Trigger will contain the value that will be tested.
- Value will contain the value that will be put in the range.

These collections compose the RDF triple. The GUI actions collection provides the predicate of the RDF triple for the GUI rules collection (see Figure 1). The GUI rules collection then instantiates each graphic component. The subject of the RDF triple can be a domain or a domain plus a trigger (the value that will be tested in the object if the predicate is some test action). The predicate corresponds to the action generated by the GUI actions collection and the object is composed of the range or the range plus a value (a value that will be placed on the range). The supplementary material includes examples that explain each of these rules and also provides the file for the project of the graphic rules (GUI rules project) with the GUI rules collection and GUI actions collection that contain data to download. They can be imported to S3DB just as in any other S3DB project.

Specifically, the web application requires the definition of up to 7 input parameters, three of which are mandatory - two are the locations for each of the two web services and the third is for an authentication token. The other four parameters will narrow the definition of the data elements on which the interface focuses and extend the range of data sources by allowing multiple deployments and users to be used. The access parameterization is extensively detailed and discussed in the documentation

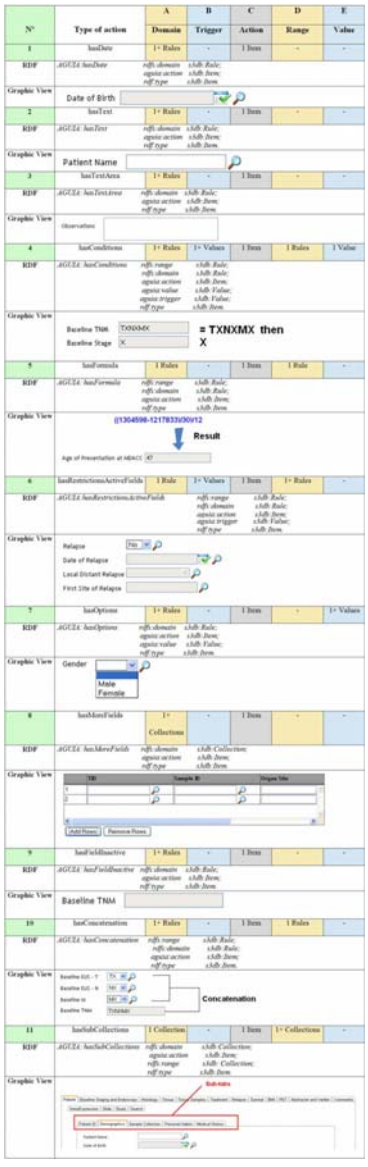


Figure 2 Detail of the 11 × 5 relationships. Details of the 11 × 5 relationships needed to configure automated assembly of the graphical user interface by the browser-based (pure JavaScript) AGUIA application. Each row corresponds to the 11 GUI actions, instantiated by S3DB items, tabulated against 5 GUI rules (columns), instantiated by S3DB rules. The 7th column is a description of the graphic component controlled by each of the 11 GUI actions. Note that S3DB is being used as a user-editable representation that is translated into standard RDF in this figure. The useful feature of S3DB as a tool, in addition to the user-editing tools, is that the distinction between domain and instantiation is always explicit [25].

provided at the open source project hosting of AGUIA (see Methods).

Both the data source and the graphic rules source are presently expected to be S3DB databases, accessed using a REST protocol, S3QL (see Materials). As discussed later, this reflects the current lack of standards for writing documents in the RDF more than it reflects a narrow focus of the S3DB prototype. The S3DB data acquisition effort described in this report was pursued in the Department of Gastrointestinal Medical Oncology at the University of Texas MD Anderson Cancer Center; hence, these data are of biomedical origin. Specifically, over a period of one year, clinical and biomedical researchers in this department submitted to an S3DB deployment a variety of clinical and biomolecular data totaling over 1 million independent S3DB "statements," which presently describe 1369 patients. These data were imported to S3DB through a script that receives the data in a tabular format. This script generates queries in S3QL, which transforms the data into S3DB RDF statements. Any S3DB project can be used to feed into the generic front-end AGUIA, although ideally AGUIA will also access a second S3DB project in which the AGUIA rules and actions are defined. The main operational advantage of relying on the S3DB web service is that the deletion, insertion and updating of data can be done through S3QL queries, whereas the reference SPARQL queries are limited to data retrieval. It is important to highlight that the data submission is completely decoupled from the configuration of the display. Consequently, it is easy to configure one or more alternative interfaces for the same data by targeting the RDF-based description of the domain (see Figure 3 for an example). More details about the data submission to S3DB can be found in the supplementary material. Figure 3 shows the ontology of the illustrative clinical trial project as a graph of user-submitted S3DB "rules" [5]. The report we reference includes graphic representations of the same RDF set using different RDF browsers that are both academic and commercial. Note in this figure that the patient ID collection plays a key role in aggregating the other collections. The supplementary material includes additional documentation and an extended discussion about how AGUIA handles the "main collection" and how the user works with this collection when performing a search, insertion or update of data.

The AGUIA can be pointed to the URI of any of the red nodes in Figure 3 (S3DB collections) to start the process of assembling the automated graphical user interface. The graphic rules come from a second data source, which, unlike the first data source, comes from a project with a predefined ontology. More specifically, this second data source for the graphic clues is a distinct S3DB web service that instantiates a fixed set of S3DB

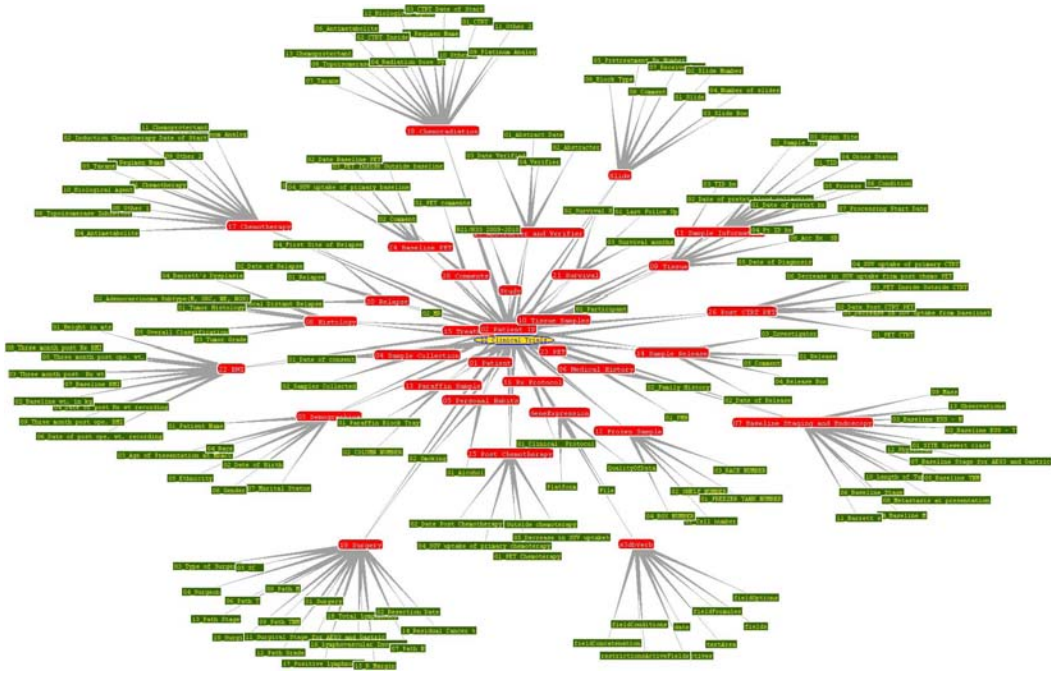


Figure 3 Ontology of clinical trials data project. User-defined ontology of the clinical trials data project in the Department of Gastrointestinal Medical Oncology, MD Anderson Cancer Center. The user describes the ontology by defining "rules," which in turn are instantiated by "statements." The oval node represents the root of the ontology; the red nodes represent rule elements that are collections of items; the green nodes represent rule objects that take literal values.

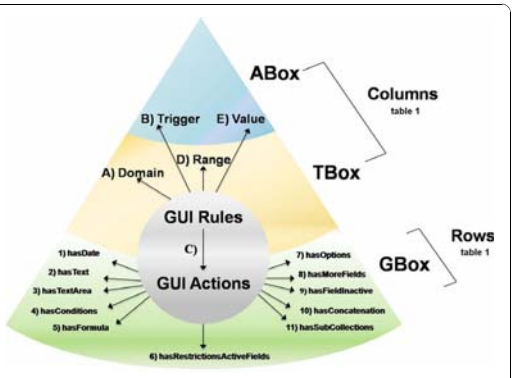


Figure 4 Graphic model of S3DB items11 × 5 S3DB rules. Graphic model as a set of 11 S3DB items (1-11) and a set of 5 S3DB rules (A-E). See Figure 2 for a translation into RDF schema by cross-tabulating the 5 GUI rules by the 11 GUI actions. The type of relationship for the GUI rules (S3DB items for the GUI actions – GBox; S3DB rules and S3DB collections – TBox; and S3DB values – ABox) identifies the type of data that instantiate each of the 16 parameters of the model.

rules (Figure 4). The RDF model proposed here was derived from those rules.

The graphic rules project (Figure 4 and Figure 2) is populated with statements that specify how to assemble the web application. In other words, by submitting statements that instantiate those rules, the user is configuring the graphical user interface as modular components. The graphic model described in Figure 4 corresponds exactly to the set of S3DB rules linking two object collections - GUI rules and GUI actions - with, respectively, 5 and 11 literal values. These rules (the 14 types of values) can be instantiated as many times as is required, and in any way needed to produce the desirable graphical appearance. This figure shows 3 parts of the knowledge base (KB): the GBox, TBox and ABox. The AGUIA GBox contains only 11 graphic components that can be used alone to compose the layout of a web application. Conventionally, the TBox indicates the terminological component, which in the AGUIA context has a domain and range that are S3DB rules, which themselves point to S3DB collections and/or S3DB rules. Finally, the ABox indicates the assertion component, which links to the context of AGUIA's

GBox as values/literals that trigger AGUIA actions (Figure 4 and Figure 2).

As highlighted in the Introduction, the identification of this model was pursued by decomposing a diversity of layouts requested by clinical researchers such that any of the layouts requested can be automatically produced by AGUIA from instantiations of this model. The translation into the RDF schema of the relationships described by the 15 S3DB rules represented in Figure 4 is the essence of the modeling work detailed in Figure 2. In this figure each RDF row represents in RDF language the instantiation of each graphic component. For example, the date of birth is always a date, therefore the specification of the nature of the data can trigger its visualization. The supplementary material also provides examples of the instantiation of each graphic component in turtle language.

The goal of the work described here was to develop a front-end web application that may be reused for different projects with different needs. The use of the AGUIA web application was illustrated in two projects with widely varying needs: a project that contains clinical/molecular data collected from patients with gastrointestinal cancer, and a workshop project that contains data collected from participants in a workshop, including their home institution, research interests and personal data. The former test case was used to produce the illustrations in this report; the latter was used for illustration in the screencast video (see the supplementary material).

JavaScript application

The AGUIA web application is a JavaScript application developed to autonomously assemble the graphical user interface using the graphic annotation clues provided by the GUI rules and actions (Figure 4 and Figure 2). This means that any project based on S3DB can be automatically accessed through the graphical user interface assembled by this web application. The assembly of the GUI starts with queries in S3QL or SPARQL that are made to the S3DB web service. The result of these queries are grouped in only one data structure. This structure contains the association between the graphic components and the components of domain ontology, containing only the information necessary to assemble the web application (Figure 5). For the two case studies the observed response times of the S3DB system to each query were 0.4 seconds on average; the response times to assemble the full web application were under 11 seconds. Figure 5 depicts the functional architecture of AGUIA through which the flow of operations can be traced. Note that the S3DB deployments independently contain both the domain ontology and the graphic model. The domain ontology describes the data content in the web application (both TBox and ABox), whereas the graphic module describes the graphic structure of the web application. Recalling from Figures 1 and 4, the latter is divided into GUI actions and GUI rules. GUI actions contain the action that can be triggered by an assertion, for example, *hasDate* will trigger *create date field*. GUI rules are divided into a domain, range, trigger and value, as described in the section *RDF protocol and list of graphic rules.* Note also that this description of the architecture is further expanded in the supplementary material. That additional documentation includes information about response times and a video with a screencast of the real-time use of AGUIA to assess the performance/response times of queries and of the web application assembly.

Note that the absence of a graphical annotation (GBox) will not prevent the application from assembling the graphical interface automatically. The sole difference is that its structure will be based on only the structure of the relationships between the data elements. Therefore, the graphic annotations provided through the GUI rules and actions can also be thought of as a way to further direct the GUI assembly process. After it is assembled, the AGUIA is able to search, insert and update registres through S3QL and SPARQL queries that are made behind the scenes by GET HTTP calls. The supplementary material includes explanations and screencast illustrations of how to search, insert and update data using AGUIA.

The AGUIA web application is currently used by distinct groups of users in the Departments of Biostatistics, Bioinformatics and Gastrointestinal Medical Oncology at MD Anderson Cancer Center. Figure 6 depicts a snapshot of AGUIA when it is pointed to the gastric oncology clinical trial research database of the Gastrointestinal Medical Oncology Department. Both goals highlighted in the Background section for this application - the autonomous assembly of the user interface in the browser in response to the invocation of an independent GBox descriptor - were fully achieved. Consequently, the application was shown to accommodate the characteristically volatile schemas and to automatically assemble new interfaces in response to changes in the data model, without the need for additional coding. As noted in the Background, data schemas were observed to change to reflect both new research data and new researchers.

The user can test AGUIA to this database through a public login (url:http://ibl.mdanderson.org/edu username and password: public). This demonstration project describes fictitious patients but the data structure is exactly the same as that in use for gastric cancer clinical trials. The video in the supplementary material describes the login in this project and allows the viewer to see the assembly of the web application. By comparing Figure 6

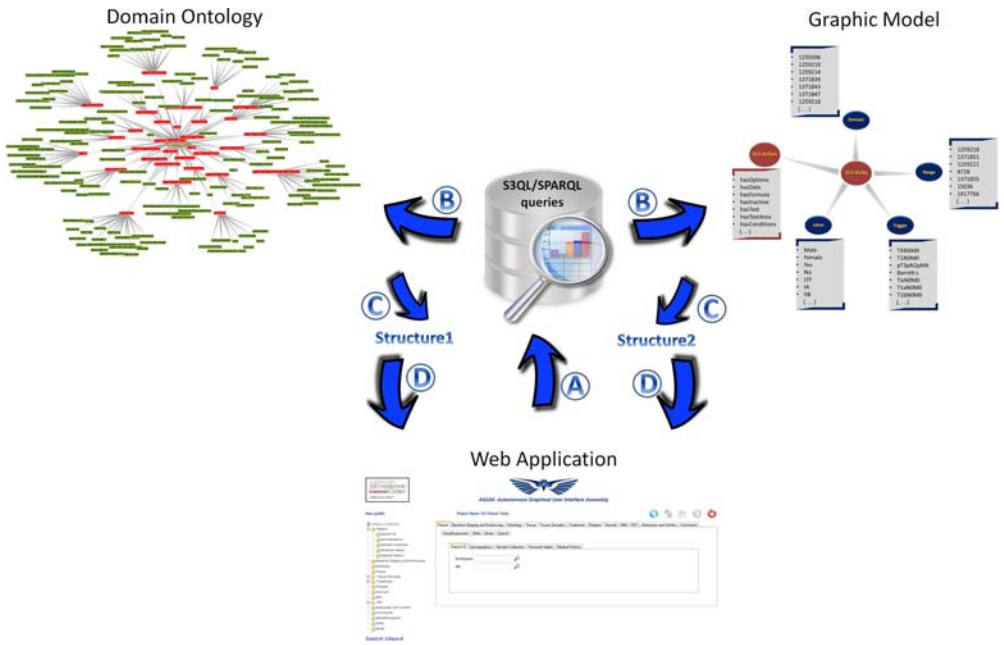


Figure 5 Automated GUI assembly process. Automated GUI assembly process: A) individual queries converted into independent URL calls in S3QL or SPARQL format; B) decomposition of the elements of the query that pertain to the domain description and to the graphic layout; C) translation of those query results into an AGUIA graphic document model (using notation 3: Project Collection Item; Rule Statement); D) merging of the two data structures back into the GUI object model, which is reflected back to the browser's own object model (DOM), as described in Figure 2.

with the ontology of the data store (Figure 3), it is apparent that a substantial number of GUI rules and GUI actions must be in use for this clinical data service. The list of S3DB rules (RDF describing the clinical domain) and the list of GUI actions and rules that modulates the assembly of the graphical user interface are provided with the supplementary material, as part of the documentation of AGUIA at the open source repository. This material includes a file in turtle/n3 format that provides the graphic rules and an exhaustive

Figure 6 Snapshot AGUIA. Snapshot of the logon screen when AGUIA is pointed to GI clinical trials S3DB web service, followed by the start page assembled by AGUIA when configured to target resource "Patient ID." Selecting another resource, say "Tissue," will produce the exact same graphic interface as if the initial target had been that resource: the operation of AGUIA is fully based on representation state transfer (REST) calls. Part A shows sub-tabs; the action used to produce this field is "hasSubCollections." Part B shows a text field; the action used to produce this field is "hasText." Part C shows a date field; the action used to produce this field is "hasDate." Part D shows a formula field; the action used to produce this field is "hasFormula." Part E shows inactive fields; the action used to produce these fields is "hasFieldInactive." Part F shows options fields; the action used to produce these fields is "hasOptions."

discussion of how each GUI action is interpreted by AGUIA. Examples of its use in the gastric oncology case study are also provided as supplementary material at the same URL.

The relevance of the assembly of the interfaces illustrated in Figure 6 in response to REST calls is that the navigation of the interface itself is a succession of REST calls. For example, the opening page was triggered by the following URL being sent to the web browser:

http://aguia.googlecode.com/hg/index.html?URLDATA=http://some_url/clinicaltrials|C1207734&KEYDATA=xxxxxxx&URLGUI=http://some_url/clinicaltrials|P1588674&KEYGUI=xxxxxxx

Starting with the domain URL, note that it targets the project hosting document directly. This illustrates an additional feature of JavaScript applications, which, being native to the web browser, remove the distinction between code hosting and application hosting - because they are the same. This parameterization of the call contains two locations and two authentications. For security reasons the actual URL and the access keys were replaced with "some_url" and "xxxxxx." The first location contains the URL of the data plus C1207734, which indicates the unique identifier of resource "PatientID." The second location contains the URL of the graphic rules plus P1588674, which indicates the instantiation of the GUI actions and rules used for the autonomous assembly. Note also that the authentication tokens, KeyDATA and KeyGUI, can be distinct, reflecting the possibility of different users seeing and configuring entirely distinct graphic interfaces for the same data.

Testing/evaluation

The test/evaluation of the illustrative case study involved 1369 patients, which corresponded to approximately 1 million independent S3DB "statements" [4], as highlighted in the RDF protocol and list of graphic rules sub-section. The test/evaluation was divided into the following steps: application assembly, search, and view, insertion and update of data.

- Application assembly

The illustrative application described in this report is assembled in about 35 seconds if all graphic components are used, and in about 29 seconds if only one of them is invoked. In either case the data store has to be reached and the GBox retrieved and processed, which accounts for most of the assembly time. After the application has been completely assembled, all subsequent actions involving rearrangements of components of the interface become nearly instantaneous because all graphic components are contained in the memory and no further consultation of the store's GBox descriptors is needed. For all practical effects, the browser-based application is equivalent to a stand-alone application except for the important fact that no "download" and "installation" steps are needed.

- Search

The search operation of the application can be simple when it involves only one node (RDF resource) and complex when it involves more than one node. In either case the search operation consists of translating the use of graphic elements assembled as unstructured by the GBox into SPARQL queries, which are then issued back to the data store. For an example query and the screencast showing the graphic operations involved in generating them, see the supplementary material at [http://sites.google.com/site/aguiadocumentation/documentation/how-to-search]. Both the single node and multiple node scenarios are analyzed hereafter by considering three scenarios of increasing complexity.

First case: Search involving only one node (one level)

In this case the search is realized directly (one level), for example, the search by participant number of all patients of the *Patient ID* collection (Figure 7).

Second case: Search involving one parent node and one child node (two levels)

In this case the search is performed at two stages (see Figure 8). In the first level the query contains the parent node (*PatientID*) and the second level contains only one child node (*Demographics*). An example of this type of search is a search by participant number and name of all the patients in the *Patient ID* and *Demographics* collections, respectively.

Third case: Search involving one parent node and two child nodes (two levels)

In this case the search is again performed at two stages (see Figure 9). For example, a search by participant number, gender and tumor grade of all patients, seeking to identify those of *female* gender with a *G2 moderately differentiated* tumor, would involve the

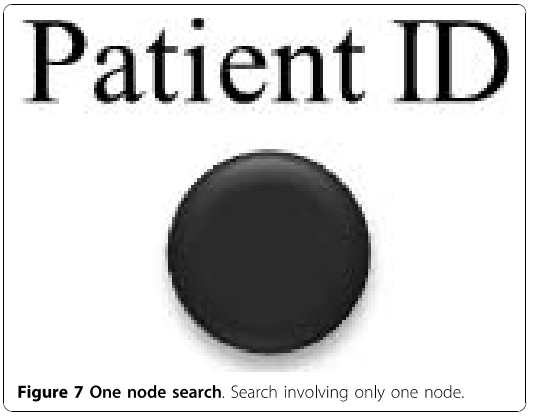


Figure 7 One node search. Search involving only one node.

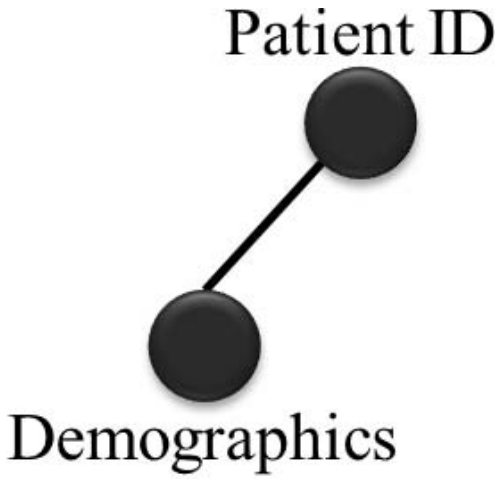


Figure 8 Two nodes search. Search involving one parent node and one child node.

Patient ID, *Demographics* and *Histology* collections, respectively.

- Execution time

The three preceding scenarios were timed to assess the performance of the application. Figure 10 illustrates the length of time required to realize a search in each one of the cases.

The results depicted in Figure 10 show that each child node added 10 seconds to the search time. The analysis of the code execution shows that this corresponds to the addition of each node to the SPARQL query, suggesting linear scaling for the query performance.

- View, insertion and update

The illustrative web application was configured to view, insert and update data one patient at a time. This takes place in one of two possible ways: View, insert and update patient data to only one node (only one collection on S3DB); or view, insert and update all patient

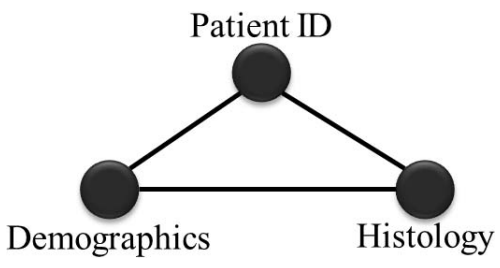


Figure 9 Three nodes search. Search involving one parent node and two child nodes.

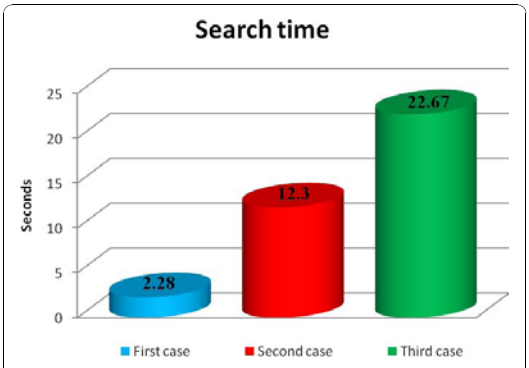


Figure 10 Search time. Search times for the three cases.

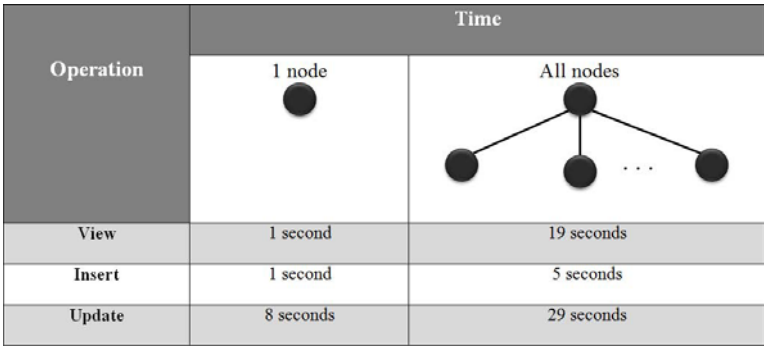
data to all nodes (all collections on S3DB). Figure 11 describes these two scenarios, including the time associated with each operation in the context of the illustrative clinical trial application and a data store of 1 million statements. As reflected by those values, the process of data store indexing is the point at which semantic databases are comparatively slow.

The performance and operation of the prototype can be seen in a screencast video in the supplementary material (http://sites.google.com/site/aguiadocumentation/video). This video shows a typical usage of the AGUIA interface.

Discussion

The knowledge base [24] defined by RDF does not, *per se*, make distinctions between assertions (ABox) and terminology (TBox) components. In other words, RDF does not by itself differentiate between what is a domain and what is an instantiation of knowledge. This is one of the three reasons why we resorted to S3DB as a mediator [4] in the creation and management of the RDF. The distinction between TBox and ABox triples corresponds, respectively, to S3DB rules and S3DB statements [5,25]. The second reason for resorting to this data service application for the study described here is the convenience of its REST API, which bodes well with the intent to produce an application that is native to the web-browser environment and which therefore can be easily used by a wide variety of domain experts in a number of computational environments. The last reason is that this work is precisely configured to test the hypothesis than an editable schema and streamlined interoperability do make a difference [3].

The main limitation of the study is the restriction of only accepting an RDF that already distinguishes between the ABox and TBox. Furthermore, and with regard to the AGUIA ontology itself, the user cannot assert types of actions beyond the 11 that are listed in Figure 2, even if it



Operation	Time	
	1 node	All nodes
View	1 second	19 seconds
Insert	1 second	5 seconds
Update	8 seconds	29 seconds

Figure 11 Execution time to realize view, insert and update. Time required to realize each one of the operations (view, insert and update), with one or all nodes.

is clear that a higher level abstraction would make a more generic use of the browser's document object model (DOM) native methods. The strengths of AGUIA when compared to other applications, such as OntoWebber [15] and WebML [14], include AGUIA's ability to assemble multiple layouts to reflect the multiple contexts of various users. In AGUIA, the user interface is more than a presentation of the page resources, it specifically responds to the semantics behind the information that is represented. As a consequence, when compared to an IIPS study [2], for example, an AGUIA browser-based application is not just more nimble but it is also simpler to use. When using IIPS, the user has a set of tools and needs to know how to use each one to get a satisfactory result in the assembly site. We argue that the literal and semantic attributes of the data can automatically drive that selection.

The AGUIA web application described in this report works with the RDF produced by the S3DB web service. Furthermore, the AGUIA graphic model, composed of combinations of GUI actions and GUI rules (Figure 4), is itself described by the RDF schema (Figure 2). Therefore, the only fundamental barrier to its application in the much wider RDF world in general is the identification of procedures that automatically annotate knowledge bases represented in that format so that they recognize its terminology and assertion components. It could be argued that RDF representations are inherently assertive [26], which poses a fundamental obstacle to its use in the elaboration of terminology. However, it is also clear that the view of the semantic web as a "web of linked data" [27] is particularly conducive to the universal aggregation required by the systems nature of both biological processes and biomedical infrastructures. Our experience with the GI clinical trials initiative described herein and also with earlier work with lung cancer research data [5] is that the accumulation of diverse data sources renders the emergence of comprehensive terminology irresistible.

It is therefore the expectation of the authors that in a domain as fluid as the life sciences in general and molecular biology in particular [3], data processing applications will have to include an iterative tool that will allow domain experts to experiment with the annotation while inspecting a self-assembled graphical user interface. If that direction is to be taken, then the graphic model we have described could be conceived as an integral component of the knowledge base - a GBox [13]. Motik proposed "GBox" to indicate the graphic specification aspect of a knowledge base, as added to the conventional ABox and TBox distinctions. Accordingly, the layout of a GUI web application is conceivably totally describable as a formal GBox component of a knowledge base. The GBox, as shown in Figure 4, was reduced to 11 or fewer components. AGUIA demonstrates how these components alone can determine how the web application was assembled (Figure 6). Such a three-component paradigm would argue that the distinction between terminology and assertion will both have an effect and be a function of the graphic presentation. It would also follow that because different domain experts request different graphic representations (GBoxes) of the knowledge base, they may also be indirectly stating that they place the boundaries between the ABox and TBox differently.

The AGUIA application allows the clinical/domain user's requests and their replies to be automatically translated into graphic interfaces. The main challenge now is to automate as much as possible the processing of RDF "spaghetti" in order to distinguish domain from instantiation such that the automated interface assembly may respond sensibly. We are aware that this is a major challenge since the Web and the content it hosts is

assertive (ABox). Accordingly, the main contribution of AGUIA will be to assist in isolating the automatic generation of a TBox that can be interactively checked by domain experts precisely because its automated graphic rendering is in place. The rendering process itself is greatly facilitated by recent improvements in the browser's graphic capabilities, particularly after the introduction of HTML5 and XForms (W3C), and by the open source libraries for user interface assembly that make use of them such as Orbeon [28]. In summary, AGUIA provides a formal, automatable bridge between RDF documents and the browser's DOM-centric extensible syntax.

Conclusions

This paper describes a web application that automatically assembles user interfaces for databases that are able to generate RDF documents that distinguish between ABox and TBox components. The tools used in this application anticipate the maturation of technologies that either have been recently developed or are still at an incipient stage of development. An example of the former is the strict use of JavaScript to develop the application such that it resides entirely on the web browser. This anticipates a trend toward using server-side components of computational environments as a representation omnibus. An example of the latter is the use of W3C's resource description framework (RDF) as read/write representation media. Currently, the SPARQL query language specifies only the read operation format. In anticipation of the write component being similarly standardized in the future, we have used a research prototype, S3DB, which allows both read and write operations on RDF-like representations.

By developing the autonomously assembled interface applications in response to specific requests from various users who were interacting with a multiplicity of domains and platforms while working with gastrointestinal clinical trials, a number of conclusions became apparent. In regard to the identification of user-friendly, domain-aware interfaces, it appears that it is more effective to develop graphic annotations before settling for a rigid distinction between assertion and terminology, which is in contrast to the more conventional approach to ontology modeling. In regard to the challenge of deploying the applications themselves, it became apparent that modularizing the interface assembly using REST protocols is particularly effective because it does not require a distinction between the universal resource identifiers (URI) that target data elements and those that configure the assembly of the graphical user interface. In conclusion, the long standing artificial intelligence (AI) challenge of contextually aware interfacing appears to benefit from the same RDF-based collaborative annotation that is behind *The Web of Linked Data*. The data-driven user annotation of graphic rules (GBox) was observed to benefit the automation of the graphic interfaces used to interact with those same data elements.

Abbreviations

AGUIA: Autonomous Graphical User Interface Assembly; RDF: Resource Description Framework; S3DB: Simple Sloppy Semantic Database; HTML: HyperText Markup Language; DHTML: Dynamic HyperText Markup Language; DOM: Document Object Model; CSS: Cascading Style Sheets; PHP: Hypertext PreProcessor; REST: Representational State Transfer; S3QL: Simple Sloppy Semantic Query Language; URI: Universal Resource Identifiers; GUI: Graphical User Interface; TBox: Terminological Component; ABox: Assertion Component; API: Application Programming Interface; GBox: Graphic Component. CTSDS: Clinical Trials Semantic Data Services; W3C: World Wide Web Consortium.

Acknowledgements

MCC acknowledges support by the CAPES Foundation (Brazil) award CAPES/LNCC 31036015, CNPQ and FAPERJ. This work was also supported in part by the Center for Clinical and Translational Sciences under contract 1UL1RR024148 from the NIH (CTSA).

Author details

[1]Department of Bioinformatics and Computational Biology, The University of Texas MD Anderson Cancer Center, 1515 Holcombe Blvd, Houston, TX 77030, USA. [2]Bioinformatics Laboratory, Laboratório Nacional de Computação Científica, Av Getúlio Vargas, 333 Petrópolis, Rio de Janeiro, Brazil. [3]Institute of Chemical and Biological Technology, Universidade Nova de Lisboa, Oeiras, Portugal. [4]Instituto Nacional de Metrologia, INMETRO, Av. Nossa Senhora das Graças 50 -prédio 6, 25250-020, Xerém - Duque de Caxias, Rio de Janeiro - Brazil. [5]Department of Gastrointestinal Medical Oncology, The University of Texas MD Anderson Cancer Center, 1515 Holcombe Blvd, Houston, TX 77030, USA. [6]The University of Texas MD Anderson Cancer Center, 1515 Holcombe Blvd, Houston, TX 77030, USA. [7]The University of Texas Medical School, 6431 Fannin St, Houston, TX 77030, USA.

Authors' contributions

MCC participated in the design and developed the application, identified the data model, and drafted the manuscript; HFD participated in modeling the database; HFD assisted with the S3DB web service; YA, SVP and JAA provided clinical data and domain expertise; ATV and JSA supervised the project and improved the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Received: 2 February 2010 Accepted: 26 October 2010
Published: 26 October 2010

References

1. Freitas F, Schulz S: **Ontologies, semantic Web and health.** *RECIIS* 2009, **3(1)**.
2. Lei Y, Motta E, Domingue J: **IIPS: an intelligent information presentation system.** *Proceedings of the 7th international conference on Intelligent user interfaces: 13-16 January 2002* .
3. Almeida JS, Chen C, Gorlitsky R, Stanislaus R, Aires-de-Souza M, Eleuterio P, Carrico J, Maretzek A, Bohn A, Chang A, Zhang F, Mitra R, Mills GB, Wang X, Deus HF: **Data integration gets 'Sloppy'.** *Nature biotechnology* 2006, **24(9)**:1070-1071.
4. Almeida JS, Deus HF, Maass W: **S3DB core: a framework for RDF generation and management in bioinformatics infrastructures.** *BMC Bioinformatics* 2010, **11**:387.
5. Deus HF, Stanislaus R, Veiga DF, Behrens C, Wistuba II, Minna JD, Garner HR, Swisher SG, Roth JA, Correa AM, Broom B, Coombes K, Chang A, Vogel LH, Almeida JS: **A Semantic Web Management Model for Integrative Biomedical Informatics.** *PLoS ONE* 2008, **3(8)**:e2946.
6. **Semantic Data Services for: Enterprise Data Interoperability.** [http://xml.gov/documents/completed/metamatrix/semanticdataservices.htm].

7. RDF Vocabulary Description Language 1.0: RDF Schema. [http://www.w3.org/TR/rdf-schema/].
8. SKOS Simple Knowledge Organization System Reference. [http://www.w3.org/TR/2009/REC-skos-reference-20090818/].
9. SKOS Simple Knowledge Organization System - Home Page. [http://www.w3.org/2004/02/skos/].
10. Hendler J, Allemang D: *Semantic Web for the Working Ontology – Effective Modeling in RDFS and OWL* Burlinghton: Morgan Kaufmann; 2008.
11. Brodie ML: Data Integration at Scale: From Relational Data Integration to Information Ecosystems. *24th IEEE International Conference on Advanced Information Networking and Applications* 2010, **2010**:2-3.
12. Giacomo GD, Lenzerini M: TBox and ABox Reasoning in Expressive Description Logics. *Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning (KR'96)* 1996, **1996**:316-327.
13. Motik B: Combining Description Logics, Description Graphs, and Rules. Springer Berlin/Heidelberg; 2009:**5749**:43-67.
14. Stefano Ceri, Piero Fraternali, Aldo Bongio: Web Modeling Language (WebML): a modeling language for designing Web sites. Elsevier North-Holland, Inc; 2000:**33**:137-157.
15. Yuhui J, Stefan D, Gio W: OntoWebber: Model-Driven Ontology-Based Web Site Management. *Proceedings of SWWS'01, The first Semantic Web Working Symposium: July 30 - August 1, 2001* .
16. DHTML Introduction. [http://www.w3schools.com/dhtml/dhtml_intro.asp].
17. DHTMLGOODIES: A library of DHTML and Ajax scripts. [http://www.dhtmlgoodies.com/].
18. Google code. [http://code.google.com].
19. Principled Design of the Modern Web Architecture. [http://www.ics.uci.edu/~taylor/documents/2002-REST-TOIT.pdf].
20. S3DB. [http://s3db.org].
21. Segaran T, Evans C, Taylor J: *Programming the Semantic Web* Sebastopol: O'Reilly Media; 2009.
22. S3DB SPARQL endpoint. [http://sparql.s3db.org/].
23. Deus HF, Veiga DF, Freire PR, Weinstein JN, Mills GB, Almeida JS: Exposing the cancer genome atlas as a SPARQL endpoint. *Journal of Biomedical Informatics* 2010, 1532-0480.
24. Donini FM, Lenzerini M, Nardi D, Schaerf A: Reasoning in description logics. *Center for the Study of Language and Information* 1997, 191-236.
25. Wang X, Gorlitsky R, Almeida JS: From XML to RDF: how semantic web technologies will change the design of 'omic' standards. *Nature biotechnology* 2005, **23(9)**:1099-1103.
26. Semantic Web Road map. [http://www.w3.org/DesignIssues/Semantic.html].
27. Linked Data-Design Issues. [http://www.w3.org/DesignIssues/LinkedData.html].
28. Orbeon. [http://www.orbeon.com/].

Pre-publication history
The pre-publication history for this paper can be accessed here:
http://www.biomedcentral.com/1472-6947/10/65/prepub

doi:10.1186/1472-6947-10-65
Cite this article as: Correa *et al.*: AGUIA: autonomous graphical user interface assembly for clinical trials semantic data services. *BMC Medical Informatics and Decision Making* 2010 **10**:65.

Chapter 4 | A domain specific language for the Semantic Web

Chapter Outline

Section 1. H.F. Deus, M.C. Correa, R. Stanislaus, M. Miragaia, W. Maass, H. de Lencastre and JS Almeida, "**S3QL: A distributed domain specific language for controlled semantic integration of life science data**" (under revision at *BMC bioinformatics*) *

The final component of this work is the identification of a domain specific language, S3QL (S3DB Query Language), targeted at aiding application developers make use of Linked Data standards to securely manage and integrate public and private biomedical data. Reflecting the data-driven nature of our approach, S3QL has been implemented as an Application Programming Interface for S3DB systems to perform controlled read/write operations on entities of the S3DB core model. In the methodology report included in this chapter, S3QL was used to effectively integrate sensitive data from three biomedical domains using RDF and SPARQL and was shown to be extensible beyond the S3DB core model and applicable also to another core model, that of the Simple Knowledge Organization System (SKOS).

* The candidate identified S3QL, implemented it as an application programming interface for S3DB, validated it with the biological use cases and wrote the manuscript

S3QL: A distributed domain specific language for controlled semantic integration of life sciences data

Helena F Deus [1,2]*, Miriã Coelho Correa[3], Romesh Stanislaus [4], Maria Miragaia[5], Wolfgang Maass [6], Hermínia de Lencastre[5,7], Ronan Fox[1] and Jonas S Almeida[8]

[1] Digital Enterprise Research Institute, National University of Ireland at Galway [2] Biomathematics, Instituto de Tecnologia Química e Biológica, Universidade Nova de Lisboa, Portugal; [3] Laboratório Nacional de Computação Científica, Petrópolis, Brasil, [4] Sanofi Pasteur, US Headquarters; [5]Laboratory of Molecular Genetics, Instituto de Tecnologia Química e Biológica, Universidade Nova de Lisboa, Av. da República, Estação Agronómica Nacional, Oeiras, Portugal; [6]Research Center for Intelligent Media, Furtwangen University, Furtwangen, Germany; [7]Laboratory of Microbiology, The Rockefeller University, New York, USA; [8]Division of Informatics, Department of Pathology, University of Alabama at Birmingham, 619 South 19th Street, Birmingham, Alamaba, USA;

* Corresponding author: helena.deus@deri.org

ABSTRACT

Introduction

The value and usefulness of data increases when it is explicitly interlinked with related data. This is the core principle of Linked Data. For life sciences researchers, harnessing the power of Linked Data to improve biological discovery is still challenged by a need to keep pace with rapidly evolving domains and requirements for collaboration and control as well as with the reference semantic web ontologies and standards. Knowledge organization systems (KOSs) can provide an abstraction for publishing biological discoveries as Linked Data without the overload of contextual descriptors.

Methods

We have previously described [1] the Simple Sloppy Semantic Database (S3DB) as an efficient model for creating knowledge organization systems using Linked Data best practices with explicit distinction between domain and instantiation and support for a permission control mechanism that automatically migrates between the two. In this report we present a domain specific language, the S3DB query language (S3QL), to operate on its underlying core model and facilitate management of Linked Data.

Results

Reflecting the data driven nature of our approach, S3QL has been implemented as an application programming interface for S3DB systems hosting biomedical data, and its syntax was subsequently generalized beyond the S3DB core model. This achievement is illustrated with the assembly of an S3QL query to manage entities from the Simple Knowledge Organization System. The illustrative use

cases include gastrointestinal clinical trials, genomic characterization of cancer by The Cancer Genome Atlas (TCGA) and molecular epidemiology of infectious diseases.

Conclusion

S3QL was found to provide a convenient mechanism to represent context for interoperation between public and private datasets hosted at biomedical research institutions and linked data formalisms.

Keywords: S3DB, Linked Data, KOS, RDF, SPARQL, knowledge organization system.

1 Background

Knowledge engineering in the Life Sciences is challenged by the combination of high specificity and high heterogeneity of the data needed to represent and understand Biology's systemic puzzles. Despite the deluge of data that has invaded life sciences in the past decade [1], data-driven discovery in Biology is hindered by a lack of enough interlinked information to allow statistical algorithms to find the patterns that inform hypothesis-driven research [3, 4]. Life Sciences research relies heavily on bioinformatics integration tools like Ensembl [5], the UCSC genome browser [6], Entrez Gene [7] or the gene ontology [8] because these offer researchers portals to a wealth of interlinked biological annotations within the context of their experimentally derived results, thus playing a lead role in advancing scientific discovery. The amount of time and effort required to develop and maintain such tools has prompted Linked Data approaches for data integration to become increasingly relevant in Health Care and Life Sciences (HCLS) domains [9-11]. Briefly stated, Linked Data can be described as a bottom-up solution for data integration: its focus is on creating a global Web of Data where typed links between data sources provide rich context and expressive reusable queries over aggregated and distributed heterogeneous datasets [9, 12-14]. The architecture of that Web is expected by its original architects [15] to require a representation of usage contexts that can be applied in the collaboration and controlled sharing of data. When this functionality is supported, as that report anticipates, “social machines” will be able to manage the simultaneous and conflicting views of data that fuel scientific debate. The S3DB knowledge organization system was designed to provide minimal support for that bottom-up process by addressing a recurrent need for controlled sharing of HCLS datasets [38, 39]. This report describes a convention, the S3QL language, to query and manipulate it. It will also be demonstrated that S3QL provides a convenient mechanism to engage Linked Data in general.

1.1 Linked Data Best Practices

Linked data best practices set the stage for an interlingua of relational data and logic in the web [16] by the definition of core principles that can be summarized as: 1) information resources should be identified with HTTP universal resource identifiers (URIs); 2) information should be served against a URI in a standard semantic web format such as the Resource Description Framework (RDF) and 3)

links should be established to information resources elsewhere [11]. For large datasets, it is also convenient that a web service supporting Sparql, the protocol and RDF and query language, is also deployed [17]. Aggregation of data sources is available either by accessing metadata about the datasets as RDF [18, 19], or through direct aggregation of RDF assertions in a single knowledgebase [20, 21]. To ensure contextual consistency and reusability across datasets, data elements and descriptors are mapped using standard vocabularies, namespaces and ontologies [22-24].

1.2. Challenges involved in Publishing Primary Experimental Life Sciences Datasets as Linked Data

The value of linked data for life scientists lies primarily in the possibility to quickly discover information about proteins or genes of interest derived, for example, from a microarray or protein array experiment [25]. Life scientists involved in primary research still face significant challenges in harnessing the power of Linked Data to improve biological discovery. Part of the difficulty lies in the lack of adequate and user-friendly mechanisms to publish biological results as Linked Data prior to publication in scientific articles. Efforts in linking life sciences data typically focus on datasets which are already available in structured and annotated formats, i.e. after the researchers have analyzed, correlated and manually annotated their results by browsing the literature or submitting their data to multiple web-based interfaces [26, 27]. Current research [28, 29] and our own experience in developing content management systems for health care and life sciences [25, 30, 31], has identified the need to go beyond those data sets by creating mechanisms for contextualizing linked life sciences data with attribution and version before it can be shared with a stable annotation. Advances have been made in that direction by other research efforts such as the recent publication of VoiD as a W3C note [32].

The technological advancements that will make primary Life Sciences experimental results an integral part of the Web of Data are also thwarted by challenges which go beyond infrastructure and standards [29]. In particular, HCLS datasets often include data elements, such as those that could be used to identify individual patients, with stringent requirements for privacy and protection [33]. The typical approach to privatizing data has been to make it the responsibility of the data providers. Although this may provide a temporary solution for a small number of self-contained datasets, it quickly becomes unmanageable when datasets aggregate both public and sensitive data from multiple sources, each with its own requirements for privacy and access control [34].

One final common concern in Life Sciences is the need to enable data experts to edit and augment the data representation models; failure to support this flexibility has lead in the past to misinterpretation of primary experimental data due to absence of critical contextual information [35-37].

1.3. Knowledge organization systems for Linked Data

In order to address the information management needs of Life Scientists, the practice of Linked Data standards must be coupled with the implementation of Knowledge Organization Systems (KOSs), a view also espoused by the W3C, where the Simple Knowledge Organization System (SKOS) has been recently proposed as a standard [38, 39]. In previous work we proposed the design principles of a KOS, the Simple Sloppy Semantic Database (S3DB) [1, 40, 41]. The S3DB core model is, much like SKOS,

task-independent and light-weight. Implementation of the S3DB Core Model and operators resulted in a prototype that has been validated and tested by Life Scientists to address pressing data management needs or, in other words, as a controlled Read-Write Linked Data system [30, 42-44]. S3DB was shown to include the minimum set of features required to support the management of experimental and analytic results by Life Sciences experts while making use of Linked Data best practices such as HTTP URI, subject-predicate-object triples represented using RDF Schema, links to widely used ontologies suggested by NCBO ontology widgets [45] or new OWL classes created by the users and a SPARQL endpoint [43]. Although complying with these practices is enough to cover the immediate query or "read" requirements of a Linked Data KOS, we found that efficient data management or "write" operations, such as inserting, updating and deprecating data instances within a KOS could be more efficiently addressed with the identification of a Domain Specific Language (DSL) to abstract most of the details involved in managing interlinked, contextualized, RDF statements.

DSLs can bridge the gap between the formalisms required by Linked Data best practices such as SPARQL and RDFS, and the basic controlled read/write management requirements of HCLS experts [13, 46, 47]. DSLs supersede general purpose languages in the identification of the domain in which a task belongs, drastically reducing the development time [48]. The task of adding a graph to a triple store is supported by most graph stores by means of the SPARQL 1.1. Update language [49]. To enable controlled "write" operations targeting the dataset, it would be useful to annotate, for example, the creator of a named graph, under which circumstances it was created and who has permission to modify it. Similarly, upon changes to the dataset, annotation of the modifier and a comment describing the change would be in the interest of the communities using the data. Many triple stores are in fact quad stores to enable partial support of that requirement for contextual representation. The most common approach is to use a named graph, a set of triples identified by a URI [50] that indicates the source of a graph. The S3DB Query Language (S3QL) presented in this report was devised with the intent of automating Linked Data management by creating those contextual descriptors in a single S3QL transaction, including author, creation date and description of the data.

By making use of those contextual descriptors, we propose a method for fine grained permission control in S3QL that relies on *s3db:operators* [51], a class of functions, with states, that may be used as the predicate of an RDF triple between a user and a dataset with privacy requirements. These operators, described in [40], operate on the transition matrix of an RDF graph and can be applied in a variety of scenarios such as optimizing queries or, as is the case with S3QL, to propagate permission assignments. We have found the target audience for S3QL to be both life sciences application developers, who use it through a RESTful application programming interface (API), and life sciences researchers who use it through user interfaces for weaving the ontologies that best represent the critical contextual information in their experimental results. The applicability of S3QL to other linked data KOSs such as the Simple Knowledge Organization System (SKOS) [39] is explored with an example and the advantages of the solution proposed are discussed in three biomedical datasets with very different requirements for controlled operations: gastrointestinal clinical trials [43], cancer genomic characterization [42] and molecular epidemiology [52].

2 Methods

This section overviews the core model for S3DB, including the set of operators that enable fine grained permission control and the distributed infrastructure supporting S3QL. The principles defined here are implemented as a prototypical application available at http://s3db.org.

2.1. The S3DB Knowledge Organization Model

S3QL is a DSL to programmatically manipulate data as instances of entities defined in a KOS. One of the key features of KOSs defined using the S3DB core model [41] is the use of typed named graphs to separate the identification of the domain, the metadata describing the data, from its observational instantiation - the data itself. We have previously shown that this approach to representing RDF greatly facilitates the assembly of SPARQL and lowers the entry barrier for biomedical researchers interested in using Semantic Web Technologies to address their data management needs [42]. That separation is achieved by using the representation of domain as triples that are themselves the predicates of the statements that instantiate that domain (as detailed by Fig. 2 in [40]). For example, the triple [*Person hasAge Age]*, identified as *:R12* through a named graph of type *s3db:rule*, describes the domain while the triple *[John :R12 26]*, identified by a named graph of type *s3db:statement*, instantiates that domain. Through the logic encoded in the RDF Schema definition of domain (rdfs:domain) and range (rdfs:range), the assertion that "John" is of type "Person" and that "26" is an "Age" is enabled in the S3DB KOS. S3DB's use of named graphs to describe the domain enables updates to the domain without affecting the consistency of its instantiation – in the example above, modifying "hasAge" with "hasAgeInYears" will not affect queries that have already been assembled using that property.

In the S3DB core [40], a meta-model for this data is also created with the specific objective of enabling propagation of operations, such as permission assignments, between the domain description and the data itself, described in the following section (see Figures 2 and 3). In the example above, the two triples are respectively assigned to entities of type *s3db:rule* and *s3db:statement* where indexes "Person" is identified by a named graph of type *s3db:collection* and "John" is identified by a named graph of type *s3db:item*. The S3DB Core specifies three other entities which are specifically devised to enable knowledge organization and operator propagation: *s3db:project* entails a list of *s3db:rule* and *s3db:collection* and are typically applied in domain contextualization; *s3db:deployment* corresponds to the physical location of an S3DB system (its URL) and *s3db:user* is the subject of permission assignment operations. It is worth noting that, by making use of S3DB entities, blank nodes are avoided by assigning a unique alphanumeric identifier to every instance of an S3DB entity. The S3DB entities can also be identified using the first letter of their names, D, P, R, C, I, S or U, which will be used in subsequent examples to indicate, respectively, *s3db:deployment*, *s3db:project*, *s3db:rule*, *s3db:collection*, *s3db:item*, *s3db:statement* or *s3db:user*.

2.2. Operators for Permission Control

The second key feature that makes S3DB appropriate for controlled management operations is support for permission control embedded in its core model. The *s3db:operators*, modulate propagation

by three core functions - merge, migrate and percolate - applied to a core model's transition matrix. This behavior for propagation of permission is described in detail in equation 5 of [40] and is reproduced here in Equation 1. The S3DB transition matrix (T) is defined by 12 *s3db:relationships* describing dependencies and inference rules between entities of the S3DB core model. The operator state vector (f) is used as the predicate of a triple established between an *s3db:user* and an entity of the S3DB core model. The JavaScript application at [53] can be used to attempt this set of propagation behaviors for *s3db:operators* with alternative transition matrixes.

$$
\begin{aligned}
& f_{object,k+1} = merge([f_{object,k}, migrate(T \times f_{subject,k})]) \\
& l = length(f) \\
& l = 1 \rightarrow migrate(f) = f = f[1] \\
& l > 1 \rightarrow migrate(f) = f[2,...,l]
\end{aligned}
\qquad \text{equation 1}
$$

The *s3db:operators* [54] have a wider scope and applicability in linked data and were put to work for permission control by defining three operator types for controlled management operations in S3QL: for each of the rights to view, change/edit or use instances of S3DB entities. The format used to assign permission was defined as a three character string, where each operator occupies respectively the first, second or third positions and may assume value N, S or Y according to the level of permission intended: no permission (N), permission limited to the creator of the resource (S) or full permission (Y). For example, the permission assignment "YSN" specifies complete permission to view (Y) the subject entity, partial permission to change it (S) and no permission to use it (N). States may be defined as dominant, by use of uppercase (Y, N or S) or recessive, by use of lowercase characters (y, n or s). Dominant and recessive permissions are used to decide on the outcome of multiple permissions converging on the same entity (as detailed in [40]). Missing permission states, indicated by the dash character '-' (which has no lower or upper case) are also allowed, as well as a mechanism to succinctly specify transitions with variable memory length (l in equation 1). The propagation of permissions in the S3DB Core Model ensures that for every entity and every user, two types of permission are defined: the assigned permission, or the permission state assigned directly to a user in an entity, and the effective permission, which is the result of the propagation of *s3db:operators*.

2.3. Components of a Distributed System

One of the requirements for RDF-based knowledge management ecosystems is the availability of queries spanning across multiple SPARQL endpoints. Automation of distributed queries in systems supporting permission control, such as S3DB, is challenged by user authentication. In S3QL, we propose addressing this through delegation to authentication authorities. As a result, a user (or usage), can be identified by a URI that is independent of the authorities that validate it. Whenever possible, it is recommended that authentication credentials be protected by use of OAuth [54].

Use of URIs to identify data elements is one of the core principles of Linked Data. However, many programming environments cannot easily handle URI as element identifiers. Problems range from decreased processing speed to a need for encoding the URIs in web service exchanges. As an anticipation for that class of problems, the URIs for entities in S3DB are interchangeable with

alphanumeric identifiers formulated as the concatenation of one of D, U, P, C, R, I or S (referring to S3DB entities described in The S3DB Knowledge Organizational Model) identifying the entity and a unique number. As an example, for a deployment located at URL http://q.s3db.org/s3dbdemo, the alphanumeric *P126* is resolvable to an entity of type *Project* with URI http://q.s3db.org/s3dbdemo/P126. To facilitate exchange of URI in distinct deployments, the URI above could also be specified as *D282:P126*, where "D282" is the alphanumeric identifier of the S3DB deployment located at URL http://q.s3db.org/s3dbdemo. Every *s3db:deployment* is identified by a named graph in the form D[number]; for completeness, metadata pertaining to each *s3db:deployment*, such as the corresponding URL, is described using the vocabulary of interlinked datasets (VoiD) [18] and shared through a root location.

2.4. Availability and Documentation

The specification of the S3QL language has been made available at http://link.s3db.org/specs and one example of the output RDF is available at http://link.s3db.org/example. S3QL has been implemented through a REST application programming interface (API) for the S3DB prototype, which is publicly available at http://s3db.org/. Both the prototype and its API were developed in PHP with MySQL or PostgreSQL for data storage. Documentation about the S3DB implementation of S3QL as an API can be found at http://link.s3db.org/docs. S3QL queries may be tested at the demo implementation at http://link.s3db.org/s3qldemo and a translator for the compact notation is available at http://link.s3db.org/translate.

3 Results

3.1. S3QL Syntax

S3QL is a domain specific language devised for facilitating management operations such as "insert", "update" or "delete" using entities of a Linked Data KOS such as the S3DB core model described above. Its syntax, however, is loosely tied to the S3DB Core Model, and can easily be applied to a set of KOSs core models in which S3DB is included. The complete syntax of S3QL in its XML (eXtended Markup Language) flavor is represented in the railroad diagram of figure 1. The S3QL syntax includes three elements: the description of the operation, the target entity and the input parameters. Four basic operation descriptions were deemed necessary to fully support read/write operations: *select*, *insert*, *update* and *delete*. The action of these operations mimic those of the structured query language (SQL) and target instances of entities (*E*) defined in the core model. Input parameters include the set of attributes defined for each of the entities either in the alphanumeric form associated with entity instances (*EntityId*) or in the form of *EntityAttributeValue* (*E.a*). The values for *E.a* are determined upon choice of *E* - for an example using S3DB entities and attributes see figure 2: *E* may be replaced with any of the entities defined in the S3DB Core Model (*Deployment*, *User*, *Project*, *Collection*, *Rule*, *Item* or *Statement*); upon choice of *E*, valid forms of *E.a* include any of the attributes defined for *E* (e.g. rdf:id, rdfs:label, rdfs:comment). A table summarizing all available operations, targets and input parameters is made available as supplementary material. The formal S3QL syntax is completed by

enclosing the outcome of one of the diagrams on figure 1 with the <S3QL> tag. For example, the following XML structure is a valid S3QL query for an operation of type *insert* where the target is the S3DB entity *Project* and the input parameter, formulated as *EntityAtributeValue* is “label=Test”:

```
<S3QL>
    <insert>project</insert>
    <where>
        <label>Test</label>
    </where>
</S3QL>
```

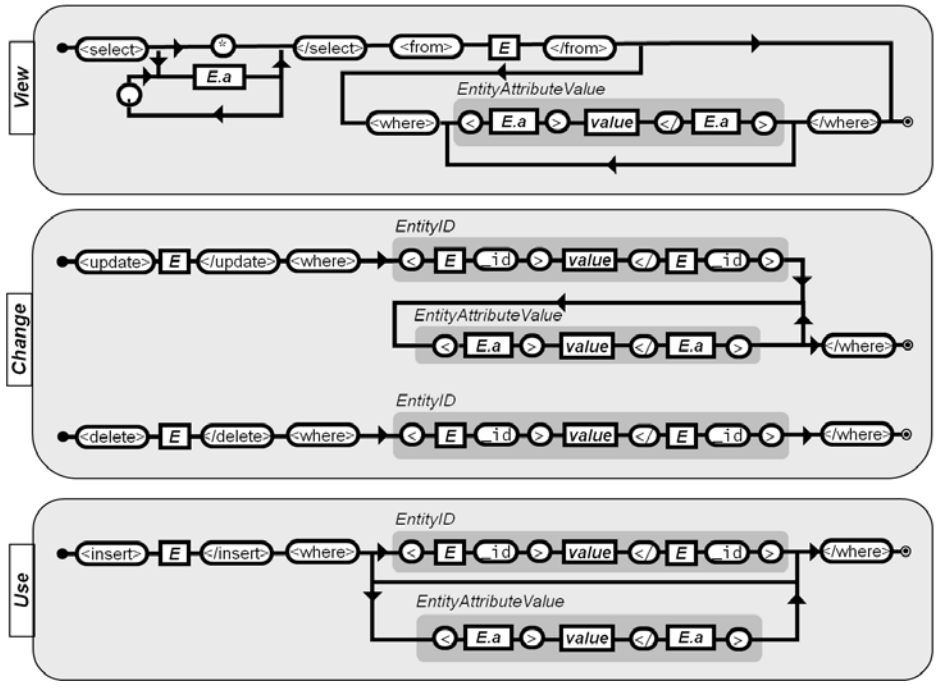


Fig. 1. S3QL language specification using rail diagrams. Rail diagrams are read from the left to the right - any string that can be composed following these diagrams is a valid S3QL query. Valid forms of *E* and *E.a* will vary according to the Core Model used in the KOS. For example, if the S3DB Core model is used, any entity in figure 2 can be used in place of *E;* upon choice of *E*, *E.a* is any attribute that can be attained by following a line from *E.*

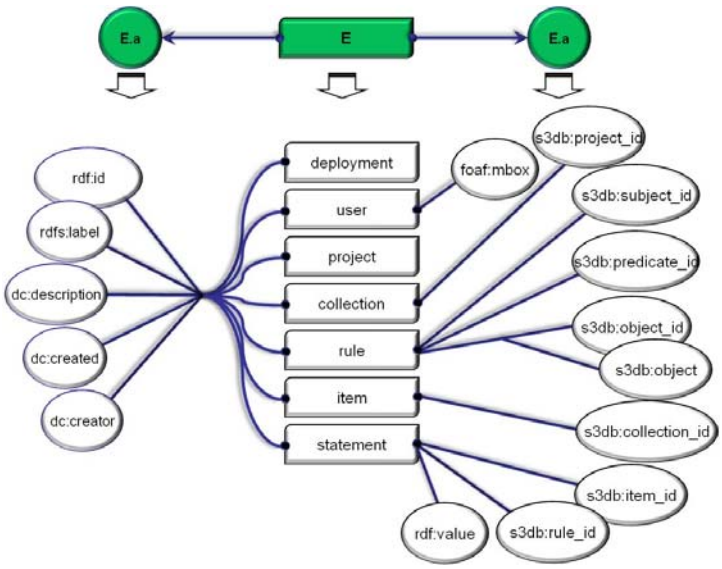


Fig. 2. Entities in the S3DB Core Model and its attributes. A minimal set of common attributes was defined (left) for each of the S3DB entities using RDF Schema (rdfs) and Dublin core (dc) terminology – these are *rdf:id*, *rdfs:label*, *dc:description*, *dc:created* and *dc:creator*. Other attributes, which are specific to each of the S3DB entities (right), such as *foaf:mbox* for the entity *User* or *s3db:project_id* for the entity *Collection* reflect the *s3db:relationships* described above and formalized in the S3DB conceptual model (figure 3).

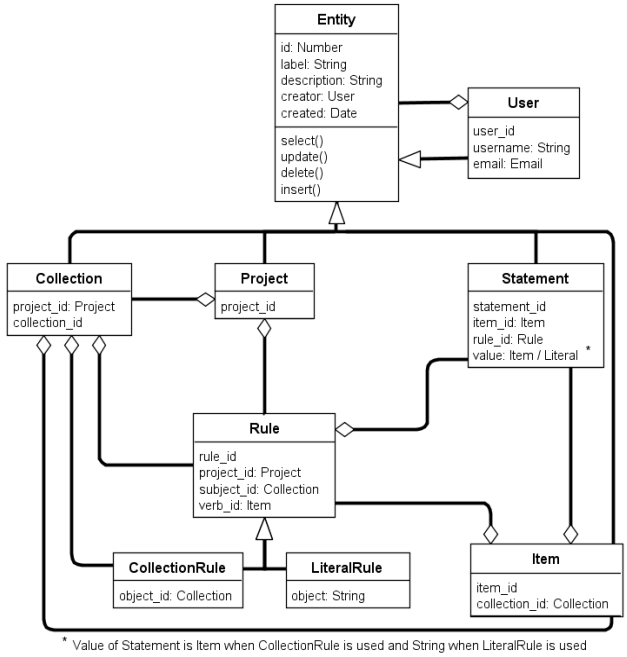


Fig 3. The S3DB conceptual model. Five attributes (*id, label, description, creator and created)* and four methods (*select, update, insert and delete*) are common to all S3DB entities. In the current S3QL implementation, the label and description attributes are defined by the submitter of the data, whereas the *id*, *created* and *creator* attributes are

automatically assigned by the system. Other dependencies were devised to comply with the definition of *s3db:relationships*.

A set of 12 *s3db:relationships* (see table 1 of [40]) determine both the direction of the permission assignment propagation and the organizational dependencies of S3DB entities. For example *s3db:PC* is the *s3db:relationship* that specifies a dependency between an instance of a *Collection* (C) and an instance of a *Project* (P) (Fig 3). The S3QL syntax fulfills this constraint by assigning *project_id* (the identifier of an S3DB *Project*) as an attribute of a *Collection*. In this description of attributes associated with the S3DB core model we make use of the assumption, as in other KOSs and in the Linked Data in general, that there is no restriction to adding relationships beyond those described here. S3QL was identified as the minimal representation to interoperate with the S3DB core model and therefore only those relationships are explored in this report.

The syntax diagram in fig. 1 generates XML, a standard widely used in web service implementations. That alternative often results in verbose queries that could easily be assembled from more compact notations. One example to consider is the form: *action (E | E.a=value)*. Here the symbol '|' should be interpreted, as in Bayesian inference, as a condition and be read "given that". The letter "E" corresponds to the first letter of any S3DB entity (D, P, R, C, I, S or U) and *E.a* is any of its attributes as described in Figure 2. In this example, the query *insert(P | label=test)* is equivalent to the example query above. That particular variant is also accepted by the S3DB prototype and a converter for this syntax into complete S3QL/XML syntax was made available at http://link.s3db.org/translate. For further compactness of this alternative formulation, entity identifiers used as parameters may be replaced with its corresponding alphanumeric identifiers - for example *project_id=156* may be replaced with *P156*. This alternative notation will be used in the subsequent examples.

3.2. S3QL Permission Control

Permission control in S3QL makes use of *s3db:operators*, which may be assigned between an *s3db:user* and a dataset with privacy concerns using an S3QL query such as *insert(U| U1,P157,permission_level=ysn),* which includes the action *insert*, the target entity *User* and three input parameters: identifier of the *User* (U1), identifier of the entity (P157) and permission assignment (ysn). Effectively, this will result in the creation of the triple *[:U1 :ysn :P157],* where the subject is of type *s3db:user*, the predicate is of type *s3db:operator* and the object is of type *s3db:project*. The inclusion of this triple in a dataset will modulate the type of management operation that a user may perform. As described in the Methods section, each position in the permission assignment operator (ysn) encodes, respectively, for permission to "view", "change" or "use" the object entity. Values y, s and n indicate, respectively, that the user has full permission to view it (y), permission to change its metadata only if he was the creator of the entity (s) and no permission to insert (n) child entities. Each S3QL operation is therefore tightly woven to each of the three operators: *select* is controlled by "view"; *update* and *delete* are controlled by "change" and insert is controlled by "use" (shaded areas in fig. 1). The 'use' operator encodes for the ability of a user to create new relationships with the target entity, which is defined separately from the right to "change" it. For example, in the case of a user (U1) being granted "y" as the effective permission to "change" an *s3db:rule*, then the metadata describing it may be altered. If, however, that same user is granted permission "n" to 'use' that same *Rule*, she is prevented from creating *Statements* using that *Rule*. Although "use" may be interpreted as being equivalent to "insert"

or “append” in other systems, we have chosen to separate the terms describing the operator “use” from the S3QL action “insert”. The permission assigned at the dataset level will then propagate in the S3DB transition matrix following the behavior formalized in equation 1, therefore avoiding the need to assign permission to every user on every entity. It is worth noting that the DSL presented here is extensible beyond the 4 management actions (select, insert, update, delete) described. The *s3db:operators* that control permission on these actions are also extensible beyond “view”, “change” and “use” and different implementations may support alternative states.

The permission control behavior for S3QL operations can be illustrated through the use of the Quadratus, a application available at http://q.s3db.org/quadratus that can be pointed at any S3DB deployment to assign permission states on S3DB entities to different users (fig. 4). Other use case scenarios are also explored in the S3QL specification at http://link.s3db.org/specs.

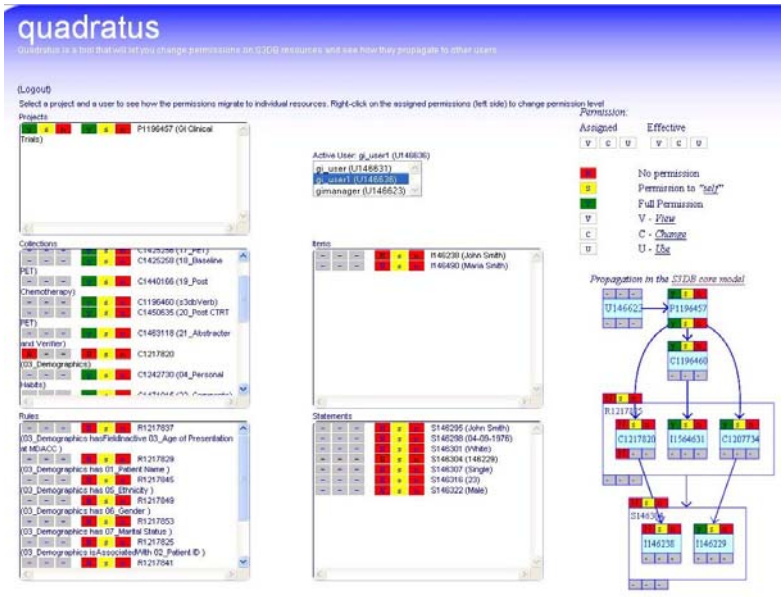


Fig. 4. Quadratus, an interface to illustrate S3QL’s permission control mechanism and its effect on S3QL queries. *Projects*, *Collections*, *Rules*, *Items* and *Statements* associated with GI Clinical Trials *Project* are displayed with effective and assigned permissions. *Collections* and *Rules* retrieved using S3QL queries *select(**C**|P1196457)* and *select(**R**| P1196457)* inherit the permission *assignment* in the *Project* “ysn“; *assigned* permission “N--” in *Collection* “Demographics” results in the *effective* permission of “Nsn” inherited by all *Rules* and *Items* that have a relationship with that *Collection*, effectively preventing gi_user1 from accessing its data. The directed labeled graph of the propagation resolution is displayed on the right side of the application illustrating the propagation mechanism.

3.3. Global Dereferencing for Distributed Queries

A simple dereferencing system was devised for S3DB identifiers that relies on the identification of root deployments, i.e. S3DB systems where alphanumeric identifiers for S3DB deployments can be dereferenced to URL. This simple mechanism enables complex transactions of controlled data. For an example of this behavior see figure 5, where the S3DB UID ***D327:**R172930*, identifying an entity of

type *Rule* (R172930) available in *deployment* **D327** is being request by a user registered in *deployment* **D309**. In order to retrieve the requested data, the URL of *deployment* **D327** must first be resolved at a root deployment such as, for example, *http://root.s3db.org*.

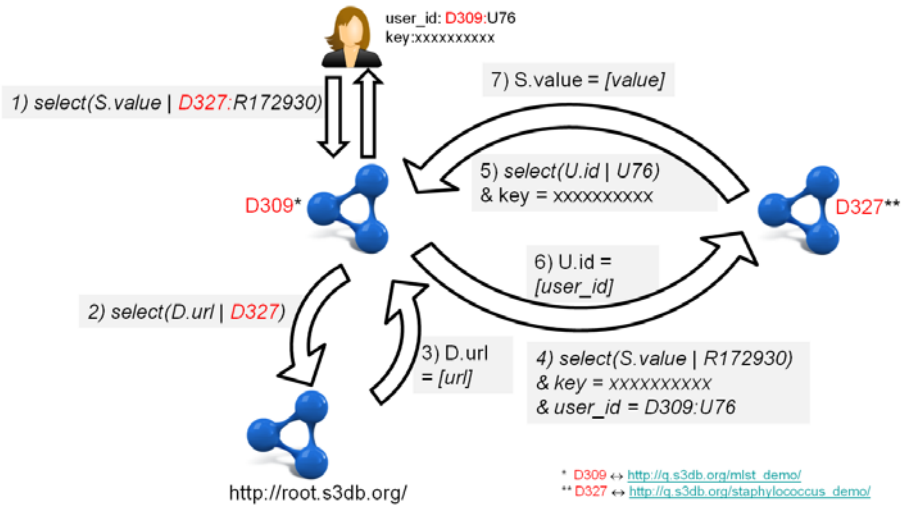


Fig. 5. The global S3QL dereferencing system. User *U78* of deployment *D309* issues a command to request all entities of type *Statement* where the attribute *Rule_id* corresponds to the value *D327:R172930* through S3QL operation *select(S.value|D327:R172930)* (1). If the URL corresponding to *deployment D327* is not cached locally or has not been validated in the past 24h, a query is issued and executed at the root deployment to retrieve the corresponding URL *select(D.url|D327)* (2,3). Once the URL is returned, query (1) is re-issued as *select(S.value| R172930)* and executed at the URL for *D327* (4). To validate the user, deployment *D327* issues the command *select(U.id|U76)* at D309 using the key provided (5,6) and returns the data only if *U.id* matches the value for *user_id* (7).

The dereferencing mechanism is also applicable in more complex cases where the root of two *deployments* sharing data is not the same. Prepending the *deployment* identifier of the root to the UIDs such as, for example, *D1016666:**D327:**R172930*, where ***D1016666*** identifies the root deployment, would result in recursive URL resolution steps such as *select(D.url| D1016666)* prior to step 4 in figure 5. This mechanism avoids broken links when S3DB *deployments* are moved to different URLs by enabling deployment metadata to be updated securely at the root using a public/private key encryption system.

3.4. Implementation and benchmarking

In the current prototype implementation, S3QL is submitted to S3DB deployments using either a GET or POST request and may include an optional authentication token (key). The REST specification [55] suggest separate HTTP methods according to the intention of the operation: often, “GET” is used to retrieve data, “PUT” is used to submit data, “POST” is used to update data and “DELETE” is used to remove data. There are, however, many programming environments that implement only the REST “GET” method, including many popular computational statistics programming frameworks such as R

and Matlab. Therefore, in order to fully explore the integrative potential of this read/write semantic web service, and to support operations beyond the 4 implemented, the S3DB prototype implementation of S3QL supports the "GET" method for all S3QL operations, with the parameters of the S3QL call appended to the URL. One drawback of relying on GET is the limits imposed by the browser on URL length. To address this potential problem, the S3DB prototype also supports the use of "POST" for S3QL calls.

Two further challenges needed to be addressed in the prototypical implementation of S3QL: 1) the need for a centralized root location to support dereferencing of deployment URI when the condensed version is used (e.g. D282) and 2) distributed queries on REST systems required users to authenticate in multiple KOSs. The first challenge was addressed by configuring an S3BD deployment as the root location, available at http://root.s3db.org. Deployment metadata is submitted to this root deployment at configuration time using S3QL; data pertaining to each deployment can therefore be dereferenced to a URL using http://root.s3db.org/D[numeric]. The option to refer to another root deployment than the default is possible during installation. To avoid overloading these root deployments with too many requests, a local 24 hour cache of all accessed deployments is kept in each S3DB deployment using the same strategy; if the URL is cached, it will not be requested from the root deployment. To address the second challenge, each *s3db:deployment* can store any number of authentication services supporting HTTP, FTP or LDAP protocols. Once the user is authenticated, temporary surrogate tokens are issued with each query. When coupled with the user identifier in the format of a URI, these tokens effectively identify the user performing the query regardless of the S3DB deployment where the query is requested.

Screencasts illustrating processing time of data manipulation using S3QL are available at http://www.youtube.com/watch?v=2KZC6kI609s and http://www.youtube.com/watch?v=FJSYLCwBaPI .

4 Discussion

One of the major concerns in making use of Linked Data to improve health care and life sciences research is the need to ensure both the availability of contextual information about experimental datasets and the ability to protect the privacy of certain data elements which may identify an individual patient. Domain specific languages (DSL) can ease the task of managing the contextual descriptors that would be necessary to implement permission control in RDF and, by doing so, could greatly accelerate the rate of adoption of Linked Data formalisms in the life sciences communities to improve scientific discovery. We have described S3QL, a DSL to perform read/write operations on entities of the S3DB Core Model. S3QL attempts to address the requirements in linking Life Sciences dataset including both publishable and un-publishable data elements by 1) including contextual descriptors for every submitted data element and 2) making use of those descriptors to ensure permission control managed by the data experts themselves. This avoids the need to break a consolidated dataset into its public and private parts when the results are acceptable for publication.

Applying S3QL to the S3DB Core Model in a prototypical application benefited from the definition of loosely defined boundaries for RDF data that enabled propagation of permission while avoiding the need to document a relationship for each data instance, individually, and for each user. The assembly of

SPARQL queries is also facilitated by the identification of domain triples using named graphs, from the data itself [42] and can be illustrated in the application at [56], where a subset of S3QL can be readily converted into the W3C standard SPARQL. Although the prototypical implementation of S3QL fits the definition of an API for S3DB systems, it is immediately apparent that the same notation could be easily and intuitively extended to other KOSs core models. For example, pointing the tool at http://q.s3db.org/translate to a JavaScript Object Notation (JSON) representation of the SKOS core model (skos.js) instead of the default S3DB core model (s3db.js) results in a valid S3QL syntax that could easily be applied as an API for SKOS based systems, as illustrated by applying the example query *"select(C|prefLabel=animal)"* using *http://link.s3db.org/translate?core=skos.js&query=select(C|prefLabel=animal)* to retrieve SKOS concepts labeled "animal". The progress of adoption for life sciences application developers can be further smoothed by complete reliance on the REST protocol for data exchange and the availability of widely used formats such as JSON, XML or RDF/turtle.

The applicability of S3QL to life sciences domains is illustrated here with three case studies: 1) in the domain of clinical trials, a project [43] that requires collaboration between departments with different interests; 2) in the domain of the cancer genome atlas (TCGA) project, a multi-institutional effort that requires multiple authentication mechanism and sharing of data among multiple institutions [42] and 3) in the domain of molecular epidemiology, a project where non-public data stored in an S3DB deployment needs to be statistically integrated with data from a public repository. All described use cases shared one considerable requirement – the ability to include, in the same dataset, both published and unpublished results. As such, they required both the annotation of contextual descriptors of the data, enabled by S3QL, and the availability of controlled permission propagation, enabled by the S3DB model transition matrix. Future work in this effort may include the application of S3QL in a Knowledge Organization System based on SKOS terminology and the definition of a transition matrix for SKOS to enable controlled permission propagation.

4.1. Gastro-intestinal Clinical Trials Use Case

As part of collaboration with the Department of Gastrointestinal Medical Oncology at The University of Texas M.D. Anderson Cancer Center, an S3DB deployment was configured to host data from gastro-intestinal (GI) clinical trials. A schema was developed using S3DB *Collections* and *Rules* and S3QL *insert* queries were used to submit data elements as *Items* and *Statements*. Two permission propagation examples are illustrated, one of a restrictive nature and the other permissive, which can also be explored at http://q.s3db.org/quadratus (Fig 4). In this example, the simple mechanisms of propagation defined for S3QL support the complex social interaction that requires a fraction of the dataset to be shared with certain users but not with others. Contextual usage is therefore a function of the attributes of the data itself (e.g. its creator) and the user identification token that is submitted with every read/write operation.

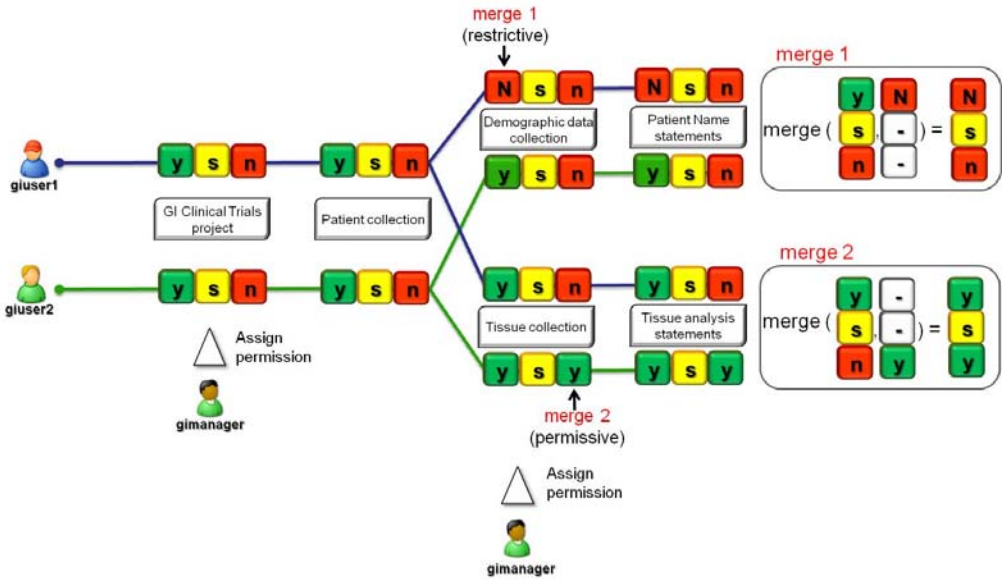


Fig 6. Two use cases of permission propagation. Two users are granted full permission to *view* GI Clinical Trials *Project* ("y"), however none of them can add new data ("n") nor edit existing data unless they were its original creator ("s"). One of the users (giuser1) is prevented from accessing any data with demographic elements such as, for example, the names of the Patients. In this case, an uppercase "N" is assigned in the right to *view* the *Collection* DemographicData, which will be merged with the inherited "y" to produce an effective permission level of "N" for the right to *view* (merge 1). For the second user (giuser2), permission is granted to *use* the *Collection* TissueData, indicating that the user can create new instances in that *Collection* (merge 2).

4.2. The Cancer Genome Atlas Use Case

The cancer genome atlas (TCGA) is a pilot project to characterize several types of cancer by sequencing and genetically characterizing tumors for over 500 patients throughout multiple institutions. S3QL was used in this case study to produce an infrastructure that exposes the public portion of the TCGA datasets as a SPARQL endpoint [42]. This was possible because SPARQL is entailed by S3QL but not the opposite. Specifically, SPARQL queries can be serialized to S3QL but the opposite is not always possible, particularly as regards write and access control operations. The structure of the S3DB Core Model which explicitly distinguishes domain from instantiation enables SPARQL query patterns, such as *?Patient :R390 ?cancerType* to be readily serialized into its S3QL equivalent *select(S|R390)*. Although this will not be further explored in this discussion, it is worth noting that the availability of this serialization allows for an intuitive syntax of SPARQL queries by patterning them on the description of the user-defined domain Rule, such as, *"Patient hasCancerType cancerType"*.

4.3. A Molecular Epidemiology Use Case

In this example, SPARQL was serialized to S3QL to support a computational statistics application. As a first step, an S3DB data store was deployed using S3QL to manage molecular epidemiology data related to strains of *Staphylococcus aureus* bacteria collected at the Instituto de Tecnologia Química e Biológica (ITQB), in Portugal. Specifically, the ITQB *Staphylococcus* reference database was devised

with a purpose of managing Multilocus Sequence Typing (MLST) data, a typing method used to track the molecular epidemiology of infectious bacteria [57-59]. As a second step, we downloaded the public *Staphylococcus aureus* MLST profiles database at http://www.mlst.net and made it available through a SPARQL endpoint at http://link.s3db.org/mlst_sparql/endpoint.php. The process of integration of MLST profiles from the ITQB *Staphylococcus* database with the publicly available MLST profiles is illustrated in fig. 7. In this example, a federated SPARQL query is assembled to access both MLST sources; data stored in the S3DB deployment is retrieved by serializing the SPARQL query into S3QL and providing an authentication token to identify the user, as described in fig. 3. The assembled graph resulting from the federated SPARQL query can be imported into a statistical computing environment such as Matlab (Mathworks Inc). Using this methodology, it was possible to cluster strains from two different data sources with very different authentication requirements. The observation that some Portuguese strains (PT1, PT2, PT15 and PT21) that are not publicly shared cluster together with a group of public UK strains (UK17,UK16,UK11,UK14,UK15,UK13,UK12) and therefore may share a common ancestor is an observation enabled by the data integrated through S3QL.

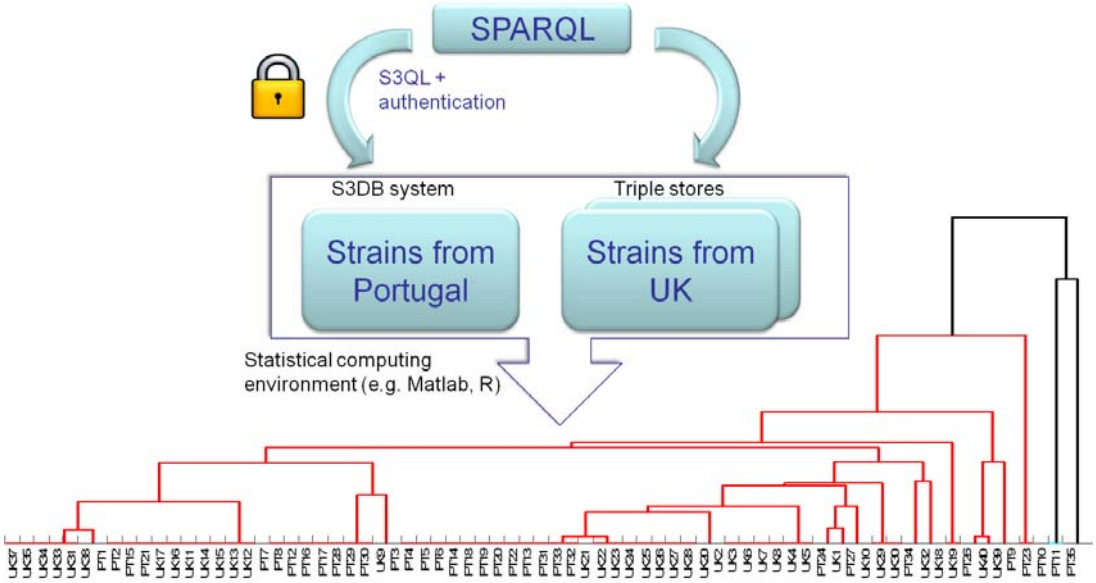


Fig. 7. Workflow for the creation of a hierarchical cluster of MLST profiles from strains collected in Portugal and in the UK. Two sources are chosen to perform the MLST profile assembly - an S3DB *deployment*, which holds the ITQB *Staphylococcus* database, and the SPARQL endpoint configured to host the *Staphylococcus aureus* MLST database. Although data in the MLST SPARQL endpoint is publicly available, to access data in S3DB, the user needs to provide an authentication token and a user_id as well as assemble the S3QL queries to retrieve the data [*select(S|R172930)* and *select(S|R167271)*], which may also be formulated as SPARQL. A data structure is assembled that can finally be analyzed using hierarchical clustering methods such as the ones made available by Matlab. The complete MLST dataset used to obtain the graph is provided as supplementary material.

5 Conclusions

Life sciences applications are set to greatly benefit from coupling Semantic Web Linked Data standards and KOSs. In the current report we illustrate data models from life sciences domains weaved using the S3DB knowledge organization system. In line with the requirements for the emergence of evolvable "social machines", different perspectives on the data are made possible by a permission propagation mechanism controlled by contextual attributes of data elements such as its creator. The operation of the S3DB KOS is mediated by the S3QL protocol described in this report, which exposes its Application Programming Interface for viewing, inserting, updating and removing data elements. Because S3QL is implemented with a distributed architecture where URIs can be dereferenced into multiple S3DB deployments, domain experts can share data on their own deployments with users of other systems, without the need for local accounts. Therefore, S3QL's fine grained permission control defined as instances of *s3db:operators* enables domain experts to clearly specify the degree of permission that a user should have on a resource and how that permission should propagate in a distributed infrastructure. This is in contrast to the conventional approach of delegating permission management to the point of access. In the current SPARQL specification extension various data sources can be queried simultaneously or sequentially. There is still no accepted mechanism for authenticating and identifying the user who issued the query, probably because SPARQL engines would have no use for that information. The critical limitation in applying this solution for Health Care and Life Sciences is the ability to make use of contextual information to determine both the level of trust on the data and to enable controlled access to elements in a dataset without breaking it and storing it in multiple systems. To address this requirement, S3QL was fitted with distributed control operational features that follow design criteria found desirable for biomedical applications. S3QL is not unique in its class, for example the linked data API, which is being used by data.gov.uk is an alternative DSL to manage linked data [60]. However, we believe that S3QL is closer to the technologies currently used by application developers and therefore may provide a more suitable middle layer between linked data formalisms and application development. It is argued that these features may assist, and anticipate, future extensions of semantic web provenance control formalisms.

6 Acknowledgements

This work was supported in part by the Center for Clinical and Translational Sciences of the Texas Medical Center at Houston under NIH (CTSA) contract no. 1UL1RR024148, by the National Cancer Institute grant 1U24CA143883-01, by the European Union FP7 PNEUMOPATH project award and by the Portuguese Science and Technology foundation under contracts PTDC/EIA-EIA/105245/2008 and PTDC/EEA-ACR/69530/2006. HFD also thankfully acknowledges PhD fellowship from the same foundation, award SFRH/BD/45963/2008 and the Science Foundation Ireland project Lion under Grant No. SFI /02/CE1/I131. The authors would also like to thank three anonymous reviewers whose comments and advice were extremely valuable for improving the manuscript.

7 Authors Contributions

HFD developed the S3QL domain specific language, implemented it in the S3DB prototype, validated it with examples and wrote the manuscript. MCC, RS, MM, HL, RF, WM and JSA tested and validated

the language with examples and made suggestions which lead to its improvement. All authors read and approved the final manuscript.

References

1. Almeida JS, Chen C, Gorlitsky R, et al. **Data integration gets "Sloppy".** *Nature biotechnology* 2006, **24**:1070-1.
2. Bell G, Hey T, Szalay A: **Computer science. Beyond the data deluge.** *Science (New York, N.Y.)* 2009, **323**:1297-8.
3. Chiang AP, Butte AJ: **Data-driven methods to discover molecular determinants of serious adverse drug events.** *Clinical pharmacology and therapeutics* 2009, **85**:259-68.
4. **The end of theory: the data deluge makes the scientific method obsolete** [http://www.wired.com/science/discoveries/magazine/16-07/pb_theory].
5. Hubbard T: **The Ensembl genome database project**. *Nucleic Acids Research* 2002, **30**:38-41.
6. Karolchik D: **The UCSC Genome Browser Database**. *Nucleic Acids Research* 2003, **31**:51-54.
7. Maglott D, Ostell J, Pruitt KD, Tatusova T: **Entrez Gene: gene-centered information at NCBI.** *Nucleic acids research* 2005, **33**:D54-8.
8. Ashburner M, Ball CA, Blake JA, et al. **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.** *Nature genetics* 2000, **25**:25-9.
9. Bizer C, Heath T, Berners-Lee T: **Linked Data - The Story So Far**. *International Journal on Semantic Web and Information Systems (IJSWIS)* 2009.
10. **Linked Data | Linked Data - Connect Distributed Data across the Web** [http://linkeddata.org/].
11. **Linked data - Design issues** [http://www.w3.org/DesignIssues/LinkedData.html].
12. Vandervalk BP, McCarthy EL, Wilkinson MD: **Moby and Moby 2: creatures of the deep (web).** *Briefings in bioinformatics* 2009, **10**:114-28.
13. **Where the semantic web stumbled, linked data will succeed - O'Reilly Radar** [http://radar.oreilly.com/2010/11/semantic-web-linked-data.html].
14. Berners-Lee T, Weitzner DJ, Hall W, et al. **A Framework for Web Science**. *Foundations and Trends® in Web Science* 2006, **1**:1-130.
15. Hendler J, Berners-Lee T: **From the Semantic Web to social machines: A research challenge for AI on the World Wide Web**. *Artificial Intelligence* 2010, **174**:156-161.
16. **Putting the Web back in Semantic Web** [http://www.w3.org/2005/Talks/1110-iswc-tbl/#(1)].
17. **SPARQL Query Language for RDF** [http://www.w3.org/TR/rdf-sparql-query].
18. Alexander K, Cyganiak R, Hausenblas M, Zhao J: **Describing Linked Datasets On the Design and Usage of voiD , the " Vocabulary Of Interlinked Datasets ."***Linked Data on the Web Workshop (LDOW 09), in conjunction with 18th International World Wide Web Conference (WWW 09)* 2009.
19. Cheung K-H, Frost HR, Marshall MS, et al. **A journey to Semantic Web query federation in the life sciences.** *BMC bioinformatics* 2009, **10 Suppl 1**:S10.
20. **A Prototype Knowledge Base for the Life Sciences** [http://www.w3.org/TR/hcls-kb/].
21. Belleau F, Nolin M-A, Tourigny N, Rigault P, Morissette J: **Bio2RDF: towards a mashup to build bioinformatics knowledge systems.** *Journal of biomedical informatics* 2008, **41**:706-16.
22. Smith B, Ashburner M, Rosse C, et al. **The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration.** *Nature biotechnology* 2007, **25**:1251-5.
23. Taylor CF, Field D, Sansone S-A, et al. **Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project.** *Nature biotechnology* 2008, **26**:889-96.
24. Noy NF, Shah NH, Whetzel PL, et al. **BioPortal: ontologies and integrated data resources at the click of a mouse.** *Nucleic acids research* 2009, **37**:W170-3.
25. Deus HF, Prud E, Zhao J, Marshall MS, Samwald M: **Provenance of Microarray Experiments for a Better Understanding of Experiment Results**. In *ISWC 2010 SWPM*. 2010.
26. Stein LD: **Integrating biological databases.** *Nature reviews. Genetics* 2003, **4**:337-45.
27. Goble C, Stevens R: **State of the nation in data integration for bioinformatics**. *Journal of Biomedical Informatics* 2008, **41**:687-693.
28. Ludäscher B, Altintas I, Bowers S, et al. **Scientific Process Automation and Workflow Management**. In *Scientific Data Management*. edited by Shoshani A, Rotem D Chapman & Hall; 2009.
29. Nelson B: **Data sharing: Empty archives.** *Nature* 2009, **461**:160-3.
30. Stanislaus R, Carey M, Deus HF, et al. **RPPAML/RIMS: a metadata format and an information management system for reverse phase protein arrays.** *BMC bioinformatics* 2008, **9**:555.
31. Silva S, Gouveia-Oliveira R, Maretzek A, et al. **EURISWEB--Web-based epidemiological surveillance of antibiotic-resistant pneumococci in day care centers.** *BMC medical informatics and decision making* 2003, **3**:9.

32. **Describing Linked Datasets with the VoiD Vocabulary** [http://www.w3.org/TR/2011/NOTE-void-20110303/].
33. **HIPAA Administrative Simplification Statute and Rules** [http://www.hhs.gov/ocr/privacy/hipaa/administrative/index.html].
34. **Socially Aware Cloud Storage** [http://www.w3.org/DesignIssues/CloudStorage.html].
35. Koslow SH: **Opinion: Sharing primary data: a threat or asset to discovery?** *Nature reviews. Neuroscience* 2002, **3**:311-3.
36. Baggerly KA, Coombes KR: **Deriving chemosensitivity from cell lines: Forensic bioinformatics and reproducible research in high-throughput biology**. *The Annals of Applied Statistics* 2009, **3**:1309-1334.
37. Kagal L, Berners-Lee T, Connolly D, Weitzner D: **Self-Describing Delegation Networks for the Web**. *Proceedings of the Seventh IEEE International Workshop on Policies for Distributed Systems and Networks* 2006:205-214.
38. Hodge G: *Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files*. 2000.
39. **SKOS Simple Knowledge Organization System Reference** [http://www.w3.org/TR/skos-reference/].
40. Almeida JS, Deus HF, Maass W: **S3DB core: a framework for RDF generation and management in bioinformatics infrastructures.** *BMC bioinformatics* 2010, **11**:387.
41. Deus HF, Stanislaus R, Veiga DF, et al. **A Semantic Web management model for integrative biomedical informatics.** *PloS one* 2008, **3**:e2946.
42. Deus HF, Veiga DF, Freire PR, et al. **Exposing The Cancer Genome Atlas as a SPARQL endpoint**. *Journal of Biomedical Informatics* 2010, **43**:998-1008.
43. Correa MC, Deus HF, Vasconcelos AT, et al. **AGUIA: autonomous graphical user interface assembly for clinical trials semantic data services.** *BMC medical informatics and decision making* 2010, **10**.
44. Freire P, Vilela M, Deus H, et al. **Exploratory analysis of the copy number alterations in glioblastoma multiforme.** *PloS one* 2008, **3**:e4076.
45. **NCBO Ontology Widgets** [http://www.bioontology.org/wiki/index.php/NCBO_Widgets].
46. Bussler C: **Is Semantic Web Technology Taking the Wrong Turn?** *Ieee Internet Computing* 2008, **12**:75-79.
47. **What people find hard about linked data** [http://dynamicorange.com/2010/11/15/what-people-find-hard-about-linked-data/].
48. Raja A, Lakshmanan D: **Domain Specific Languages**. *International Journal of Computer Applications* 2010, **1**:99–105.
49. **SPARQL Update** [http://www.w3.org/TR/sparql11-update/].
50. **Named Graphs / Semantic Web Interest Group** [http://www.w3.org/2004/03/trix/].
51. **S3DB operator function states** [http://code.google.com/p/s3db-operator/].
52. Deus HF, Sousa MA de, Carrico JA, Lencastre H de, Almeida JS: **Adapting experimental ontologies for molecular epidemiology.** In *AMIA Annual Symposium proceedings*. 2007:935.
53. **S3DB Operators** [http://s3db-operator.googlecode.com/hg/propagation.html].
54. **The OAuth 1.0 Protocol** [http://tools.ietf.org/html/rfc5849].
55. Fielding RT: **Architectural Styles and the Design of Network-based Software Architectures**. *Technology* 2000.
56. **S3QL serialization engine** [http://js.s3db.googlecode.com/hg/translate/quickTranslate.html].
57. Francisco AP, Bugalho M, Ramirez M, Carriço JA: **Global optimal eBURST analysis of multilocus typing data using a graphic matroid approach.** *BMC bioinformatics* 2009, **10**:152.
58. Ippolito G, Leone S, Lauria FN, Nicastri E, Wenzel RP: **Methicillin-resistant Staphylococcus aureus: the superbug.** *International journal of infectious diseases* 2010, **14**:S7-S11.
59. Harris SR, Feil EJ, Holden MTG, et al. **Evolution of MRSA during hospital transmission and intercontinental spread.** *Science (New York, N.Y.)* 2010, **327**:469-74.
60. **Linked Data API** [http://code.google.com/p/linked-data-api/].

Final Discussion

We have presented the design principles and validation use cases of S3DB, a Knowledge Organization System devised to address the data management requirements of biomedical domain experts. Biomedical domains have very singular properties that challenge their persistent integration and scalable representation: 1) their data models are often updated to include new experimental variables and 2) their sensitive combination of public and private data require a mechanism for fine grained permission management.

S3DB tackles the first challenge through use of Linked Data best practices, in particular by representing data as RDF triples [90, 91]. A distinctive feature of RDF is its separation of content and presentation, a feature that allows the content to fluidly evolve while the representation formalisms are rigidly kept unaltered. In practice, this means that RDF based representations, and S3DB representations by extension, tolerate changes to the data models without affecting interoperability with previous versions. As a consequence, data models in S3DB can be collaboratively weaved by the biology domain experts themselves through annotation with experimental data [90]. This feature was specifically intended to support data-driven models that accurately reflect the parameters being measured. However, a possible setback in the widespread adoption of this approach is the data modeling task, which has traditionally been the responsibility of bioinformatics experts, becoming the responsibility of biomedical researchers. The effort required to model a biological domain can be minimized by taking advantage of the portability of RDF representations - data models in S3DB can be transferred across S3DB deployments and further edited to address specific data management requirements. It is also possible to make partial use of models in other S3DB systems even when they are not publicly available. Automating this process by making shared

models or fractions thereof widely available through S3DB associated interfaces could greatly facilitate the emergence and support the collaborative evolution of domain representations and terminologies. In fact, this was already partially addressed through the availability of a centralized location at *http://root.s3db.org* where data models may be exchanged. We have illustrated in [92] that partial interoperability is often enough to answer specific cross domain queries.

To address the second challenge, a set of desirable behaviors for the propagation of assigned permissions in the distributed S3DB Core Model was identified and mathematically formalized in [93]. Permission management has not been standardized in Semantic Web formalisms and existing implementations have mostly fallen into one of two extremes: either permission is defined at the point of entry, or it becomes necessary to assign permission for every user in every data element. Reflecting our bottom-up approach, data elements in S3DB are described as attribute-value pairs. A management model for S3DB was thus identified to encapsulate those attribute-value pairs within meaningful knowledge domains without affecting integration across domains [91]. This achievement is a direct result of annotating data elements to S3DB entities and a key ingredient for supporting the fine grained permission management mechanisms that allows users of S3DB to mix both public and private data in the same model [93]. A common practical concern for systems supporting flexible propagation of elements such as the *s3db:operators* is the scalability issues that may arise from the need to re-calculate the operator states with every update. This can be solved by making extensive use of caching mechanisms, as benchmarked by the use case at [95]. Another consequence of the permission propagation mechanism is that different usages and perspectives on the same data may arise. Although this may appear contradictory to the orthogonality requirements of top-down approaches, it sets the stage for the emergence of the much more appealing “social machines” [94], community evolved overlapping

representations that act as ecosystems where researchers selectively share data and resources.

The design choice in the S3DB Core Model whereby S3DB entities are used to separate domain descriptors, or the elements describing the data, from the data itself, was found to be critical for data integration [92], visualization [95] and management [96]. In bottom-up approaches RDF assertions are often created without formally identifying the domain descriptors, which often leads to significant amounts of time “eye-parsing” the data in order to assemble meaningful SPARQL queries [97]. Top-down approaches rely instead on a combination of OWL and RDF to represent respectively the domain descriptors and its instantiations, which creates a gap between modeling and annotating the data [98]. Furthermore, it is not always possible in biomedical domains to ensure the orthogonality required for top-down approaches. The solution devised for S3DB falls somewhere between the two approaches: domain descriptors are RDF triples annotated to S3DB Rules. We have illustrated how this solution assists both in the assembly of intuitive SPARQL queries by making use of user-defined S3DB Rules as templates for the query [92] and to enable the dynamic generation of data driven intuitive interfaces [95]. The alternative solution of automating the creation of RDF triples from its native relational or XML formats results in domain descriptions that reflect the original schema rather than the structure in the data.

Because of their novelty, semantic web technologies have not yet evolved to a point where they have become as useful for biomedical researchers as the read/write Web 2.0 tools that are currently applied for data and knowledge sharing [99, 100]. S3QL, the final component of S3DB, was devised as a domain specific language targeted at lowering the bar for biomedical application developers interested in using semantic web technologies for their integrative pursuits [96]. It is not yet clear, to the Linked Data and the Semantic Web community in general, how data collection in user-devised models will meet with the logical formalisms

required for decision making and by doing so reduce the knowledge re-engineering bottleneck. Contextual representations, in which an assertion can be said to be true within a particular context or usage, is a likely solution [94]. Accordingly, S3QL operations make use of the contextual descriptors, such as the data element creator or the identity of the user issuing a query, to make decisions regarding management operations. The syntax devised for S3QL is easily extensible to other operations and even to other semantic web core models, as illustrated in [96] with the Simple Knowledge Organization System, and could indeed be extended to support inference operations.

References

1. **Science is organized knowledge. Wisdom is organized life.** [http://www.quotationspage.com/quote/26217.html].
2. Bourne PE, Beran B, Bi C, et al. **Will widgets and semantic tagging change computational biology?** *PLoS computational biology* 2010, **6**:e1000673.
3. **These Days: Science in the Information Age** [http://www.kpbs.org/news/2010/mar/01/science-information-age/].
4. Studer R: **Knowledge engineering: Principles and methods**. *Data & Knowledge Engineering* 1998, **25**:161-197.
5. Shortliffe EH, Buchanan BG, Feigenbaum EA: **Knowledge engineering for medical decision making: A review of computer-based clinical decision aids**. *Proceedings of the IEEE* 1979, **67**:1207-1224.
6. Ruttenberg A, Clark T, Bug W, et al. **Advancing translational research with the Semantic Web.** *BMC bioinformatics* 2007, **8**.
7. Drolet BC, Lorenzi NM: **Translational research: understanding the continuum from bench to bedside.** *Translational research : the journal of laboratory and clinical medicine* 2011, **157**:1-5.
8. Payne PRO, Embi PJ, Niland J: **Foundational biomedical informatics research in the clinical and translational science era: a call to action.** *Journal of the American Medical Informatics Association : JAMIA* 2010, **17**:615-6.
9. Zerhouni EA: **Translational and Clinical Science — Time for a New Vision**. *New England Journal of Medicine* 2005, **353**:1621-1623.
10. Woolf SH: **The meaning of translational research and why it matters.** *JAMA : the journal of the American Medical Association* 2008, **299**:211-3.
11. **European Commission Research & Innovation** [http://ec.europa.eu/research/fp7/index_en.cfm?pg=health].
12. Hoekstra R: **The knowledge reengineering bottleneck**. *Semantic Web* 2010, **1**:111-115.
13. Blake J a, Bult CJ: **Beyond the data deluge: data integration and bio-ontologies.** *Journal of biomedical informatics* 2006, **39**:314-20.
14. Gershon D: **Dealing with the Data Deluge**. *Nature* 2002, **416**:889-891.
15. Brooksbank C, Quackenbush J: **Data standards: a call to action.** *Omics : a journal of integrative biology* 2006, **10**:94-9.
16. Butte A: **The use and analysis of microarray data**. *Nature Reviews* 2002, **1**:951-960.
17. Kitano H: **Computational systems biology.** *Nature* 2002, **420**:206-10.
18. Kitano H: **Systems biology: a brief overview.** *Science (New York, N.Y.)* 2002, **295**:1662-4.
19. Voit EO, Brigham KL: **The Role of Systems Biology in Predictive Health and Personalized Medicine**. *The Open Pathology Journal* 2008, **2**:68-70.

20. Li P, Dada JO, Jameson D, et al. **Systematic integration of experimental data and models in systems biology.** *BMC bioinformatics* 2010, **11**:582.
21. Searls DB: **Data integration: challenges for drug discovery.** *Nature reviews. Drug discovery* 2005, **4**:45-58.
22. Pober JS, Neuhauser CS, Pober JM: **Obstacles facing translational research in academic medical centers.** *The FASEB journal : official publication of the Federation of American Societies for Experimental Biology* 2001, **15**:2303-13.
23. Lindberg DAB: **Medicine and Health on the Internet: The Good, the Bad, and the Ugly**. *JAMA: The Journal of the American Medical Association* 1998, **280**:1303-1304.
24. Altman RB, Klein TE: **Challenges for biomedical informatics and pharmacogenomics.** *Annual review of pharmacology and toxicology* 2002, **42**:113-33.
25. Goble C, Stevens R: **State of the nation in data integration for bioinformatics.** *Journal of biomedical informatics* 2008, **41**:687-93.
26. Quackenbush J: **Data standards for "omic" science.** *Nature biotechnology* 2004, **22**:613-4.
27. Taylor CF, Hermjakob H, Julian RK, et al. **The work of the Human Proteome Organisation's Proteomics Standards Initiative (HUPO PSI).** *Omics : a journal of integrative biology* 2006, **10**:145-51.
28. Brazma a, Hingamp P, Quackenbush J, et al. **Minimum information about a microarray experiment (MIAME)-toward standards for microarray data.** *Nature genetics* 2001, **29**:365-71.
29. Barrett T, Troup DB, Wilhite SE, et al. **NCBI GEO: archive for high-throughput functional genomic data.** *Nucleic acids research* 2009, **37**:D885-90.
30. Brazma A, Parkinson H, Sarkans U, et al. **ArrayExpress--a public repository for microarray gene expression data at the EBI.** *Nucleic acids research* 2003, **31**:68-71.
31. Hubbard T, Barker D, Birney E, et al. **The Ensembl genome database project.** *Nucleic acids research* 2002, **30**:38-41.
32. Karolchik D: **The UCSC Genome Browser Database**. *Nucleic Acids Research* 2003, **31**:51-54.
33. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: **Basic local alignment search tool.** *Journal of molecular biology* 1990, **215**:403-10.
34. Apweiler R, Bairoch A, Wu CH, et al. **UniProt: the Universal Protein knowledgebase.** *Nucleic acids research* 2004, **32**:D115-9.
35. Ashburner M, Ball CA, Blake JA, et al. **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.** *Nature genetics* 2000, **25**:25-9.
36. Stein LD: **Integrating biological databases.** *Nature reviews. Genetics* 2003, **4**:337-45.
37. Schonbach C, Kowalski-Saunders P, Brusic V: **Data warehousing in molecular biology**. *Brief Bioinform* 2000, **1**:190-198.

38. Stead W, Miller R, Musen M, Hersh W: **Integration and Beyond: Linking Information from Disparate Sources and into Workflow**. *Journal of the American Medical Informatics Association* 2000, **7**:135-145.
39. Louie B, Mork P, Martin-Sanchez F, Halevy A, Tarczy-Hornoch P: **Data integration and genomic medicine.** *Journal of biomedical informatics* 2007, **40**:5-16.
40. Vandervalk BP, McCarthy EL, Wilkinson MD: **Moby and Moby 2: creatures of the deep (web).** *Briefings in bioinformatics* 2009, **10**:114-28.
41. Dowell R, Jokerst R, Day A, Eddy S, Stein L: **The Distributed Annotation System**. *BMC Bioinformatics* 2001, **2**:7.
42. Lenzerini M, Sapienza L, Salaria V, Roma I-: **Data Integration : A Theoretical Perspective.** *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* 2002.
43. Chung SY, Wong L: **Kleisli: a new tool for data integration in biology.** *Trends in biotechnology* 1999, **17**:351-5.
44. Achard F, Vaysseix G, Barillot E: **XML, bioinformatics and data integration**. *Bioinformatics(Oxford, England)* 2001, **17**:115-125.
45. Hucka M: **The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models**. *Bioinformatics* 2003, **19**:524-531.
46. Stanislaus R, Chen C, Franklin J, Arthur J, Almeida JS: **AGML Central: web based gel proteomic infrastructure.** *Bioinformatics (Oxford, England)* 2005, **21**:1754-7.
47. Spellman P, Miller M, Stewart J, et al. **Design and implementation of microarray gene expression markup language (MAGE-ML)**. *Genome Biology* 2002, **3**:research0046.1-research0046.9.
48. Pedrioli PGA, Eng JK, Hubley R, et al. **A common open representation of mass spectrometry data and its application to proteomics research.** *Nature biotechnology* 2004, **22**:1459-66.
49. Wang X, Gorlitsky R, Almeida JS: **From XML to RDF: how semantic web technologies will change the design of "omic" standards.** *Nature biotechnology* 2005, **23**:1099-103.
50. Cheung K-H, Yip KY, Smith A, et al. **YeastHub: a semantic web use case for integrating data in the life sciences domain.** *Bioinformatics (Oxford, England)* 2005, **21**:i85-96.
51. Smith A, Cheung K, Krauthammer M, Schultz M, Gerstein M: **Leveraging the structure of the Semantic Web to enhance information retrieval for proteomics.** *Bioinformatics (Oxford, England)* 2007, **23**:3073-9.
52. Sahoo SS, Zeng K, Bodenreider O, Sheth A: **From "glycosyltransferase" to "congenital muscular dystrophy": integrating knowledge from NCBI Entrez Gene and the Gene Ontology.** *Studies in health technology and informatics* 2007, **129**:1260-4.
53. Kozhenkov S, Dubinina Y, Sedova M, et al. **BiologicalNetworks 2.0 - an integrative view of genome biology data.** *BMC bioinformatics* 2010, **11**:610.

54. Good BM, Tennis JT, Wilkinson MD: **Social tagging in the life sciences: characterizing a new metadata resource for bioinformatics.** *BMC bioinformatics* 2009, **10**:313.
55. Good BM, Wilkinson MD: **The Life Sciences Semantic Web is full of creeps!** *Briefings in bioinformatics* 2006, **7**:275-86.
56. **Semantic Web - W3C** [http://www.w3.org/standards/semanticweb/].
57. Berners-Lee T, Hendler J, Lassila O: **The Semantic Web**. *Scientific American* 2001, **284**:34-43.
58. Berners-Lee T, Kagal L: **The Fractal Nature of the Semantic Web**. *Artificial Intelligence Magazine* 2008, **29**:29.
59. **Linked data - Design issues** [http://www.w3.org/DesignIssues/LinkedData.html].
60. **Resource Description Framework (RDF): Concepts and Abstract Syntax** [http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/].
61. Markowitz VM: **Representing processes in the extended entity-relationship model.** *Proceedings of the Sixth International Conference on Data Engineering* 1990:103-110.
62. **SPARQL Query Language for RDF** [http://www.w3.org/TR/rdf-sparql-query].
63. Bizer C, Heath T, Berners-Lee T: **Linked Data - The Story So Far**. *International Journal on Semantic Web and Information Systems (IJSWIS)* 2009.
64. **RDF Vocabulary Description Language 1.0: RDF Schema** [http://www.w3.org/TR/rdf-schema/].
65. **OWL - Web Ontology Language Document Overview** [http://www.w3.org/TR/owl-features/].
66. Allemang D, Hendler J: *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Morgan Kaufmann; 2008:352.
67. Goble C, Wroe C: **The montagues and the capulets.** *Comparative and functional genomics* 2004, **5**:623-32.
68. Soldatova LN, King RD: **Are the current ontologies in biology good ontologies?** *Nature biotechnology* 2005, **23**:1095-8.
69. Gruber T: **A Translation Approach to Portable Ontology Specifications**. *International Journal Human-Computer Studies* 1995, **43**:907-928.
70. Sparkes A, Aubrey W, Byrne E, et al. **Towards Robot Scientists for autonomous scientific discovery.** *Automated experimentation* 2010, **2**:1.
71. Soldatova LN, Clare A, Sparkes A, King RD: **An ontology for a Robot Scientist.** *Bioinformatics (Oxford, England)* 2006, **22**:e464-71.
72. Smith B, Ashburner M, Rosse C, et al. **The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration.** *Nature biotechnology* 2007, **25**:1251-5.
73. Quackenbush J: **Top-down standards will not serve systems biology**. *Nature* 2006, **440**:24.
74. Musen M, Lewis S, Smith B: **Wrestling with SUMO and bio-ontologies**. *Nature Biotechnology* 2006, **24**:21-22.

75. Miller M, Rifaieh R: **Wrestiling with SUMO and Bio-Ontologies**. *Nature biotechnology* 2006, **24**:23.
76. Marshall MS, Prud'hommeaux E: **HCLS Knowledgebase**. 2008.
77. **HCLSIG/LODD - ESW Wiki** [http://esw.w3.org/HCLSIG/LODD].
78. Belleau F, Nolin M-A, Tourigny N, Rigault P, Morissette J: **Bio2RDF: towards a mashup to build bioinformatics knowledge systems.** *Journal of biomedical informatics* 2008, **41**:706-16.
79. Dumontier M: **Building an effective Semantic Web for health care and the life sciences**. *Semantic Web* 2010, **1**:131-135.
80. Cheung K-H, Prud'hommeaux E, Wang Y, Stephens S: **Semantic Web for Health Care and Life Sciences: a review of the state of the art.** *Briefings in bioinformatics* 2009, **10**:111-3.
81. **Linked Data Cloud diagram** [http://richard.cyganiak.de/2007/10/lod/].
82. **Sesame** [http://www.openrdf.org/].
83. Noy NF, Shah NH, Whetzel PL, et al. **BioPortal: ontologies and integrated data resources at the click of a mouse.** *Nucleic acids research* 2009, **37**:W170-3.
84. Taylor CF, Field D, Sansone S-A, et al. **Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project.** *Nature biotechnology* 2008, **26**:889-96.
85. **SKOS Simple Knowledge Organization System Reference** [http://www.w3.org/TR/skos-reference/].
86. Hendler J: **The Dark Side of the Semantic Web**. *IEEE Intelligent Systems* 2007, **22**:2-4.
87. Maglott D, Ostell J, Pruitt KD, Tatusova T: **Entrez Gene: gene-centered information at NCBI.** *Nucleic acids research* 2005, **33**:D54-8.
88. Goh K-il, Cusick ME, Valle D, et al. **The human disease network**. *PNAS* 2007, **104**:8685-8690.
89. Nardi D, Brachman RJ: **An Introduction to Description Logics.** In *The description logic handbook*. New York, New York, USA: Cambridge University Press New York; 2003:6-44.
90. Almeida JS, Chen C, Gorlitsky R, et al. **Data integration gets "Sloppy".** *Nature biotechnology* 2006, **24**:1070-1.
91. Deus HF, Stanislaus R, Veiga DF, et al. **A Semantic Web management model for integrative biomedical informatics.** *PloS one* 2008, **3**:e2946.
92. Deus HF, Veiga DF, Freire PR, et al. **Exposing The Cancer Genome Atlas as a SPARQL endpoint**. *Journal of Biomedical Informatics* 2010, **43**:998-1008.
93. Almeida JS, Deus HF, Maass W: **S3DB core: a framework for RDF generation and management in bioinformatics infrastructures.** *BMC bioinformatics* 2010, **11**:387.
94. Hendler J, Berners-Lee T: **From the Semantic Web to social machines: A research challenge for AI on the World Wide Web**. *Artificial Intelligence* 2010, **174**:156-161.
95. Correa MC, Deus HF, Vasconcelos AT, et al. **AGUIA: autonomous graphical user interface assembly for clinical trials semantic data services.** *BMC medical informatics and decision making* 2010, **10**.

96. Deus HF, Correa MC, Stanislaus R, et al. **S3QL: A distributed domain specific language for controlled semantic integration of life science data**. *BMC bioinformatics* **(under revision)**.
97. Jarrar M, Dikaikos M: **MashQL : A Query-by-Diagram Topping SPARQL**. *Proceeding of the 2nd international workshop on Ontologies and information systems for the semantic web* 2008:89-96.
98. Motik B, Horrocks I, Sattler U: **Bridging the gap between OWL and relational databases**. *Web Semantics: Science, Services and Agents on the World Wide Web* 2009, **7**:74-89.
99. Zhang Z, Cheung K-H, Townsend JP: **Bringing Web 2.0 to bioinformatics.** *Briefings in bioinformatics* 2009, **10**:1-10.
100. Neumann E: **A life science Semantic Web: are we there yet?** *Science's STKE : signal transduction knowledge environment* 2005, **2005**:pe22.

Appendix

This section includes two publications co-authored by the candidate which, although not a core part of this thesis, made use of and enabled further improvements to the S3DB prototype.

Section 1. R. Stanislaus, M. Carey, H.F. Deus, K. Coombes, B.T. Hennessy, G.B. Mills, and J.S. Almeida, "**RPPAML/RIMS: a metadata format and an information management system for reverse phase protein arrays**", *BMC bioinformatics*, vol. 9, Jan. 2008, p. 555..*

***** The candidate developed the S3DB prototype and extensions that were used to support data management, helped identify the representation model and tested the application with use cases

Section 2. P. Freire, M. Vilela, H. Deus, Y.-W. Kim, D. Koul, H. Colman, K.D. Aldape, O. Bogler, W.K.A. Yung, K. Coombes, G.B. Mills, A.T. Vasconcelos, and J.S. Almeida, "**Exploratory analysis of the copy number alterations in glioblastoma multiforme**," *PloS one*, vol. 3, Jan. 2008, p. e4076..**

** The candidate developed the S3DB prototype and helped devise the data framework that was used to access and traverse the data

BMC Bioinformatics

BioMed Central

Research article

Open Access

RPPAML/RIMS: A metadata format and an information management system for reverse phase protein arrays

Romesh Stanislaus*[1], Mark Carey[2], Helena F Deus[1,3], Kevin Coombes[1], Bryan T Hennessy[2], Gordon B Mills[2] and Jonas S Almeida[1]

Address: [1]Department of Bioinformatics and Computational Biology, The University of Texas M. D. Anderson Cancer Center, Houston, Texas, USA, [2]Department of Systems Biology, The University of Texas M. D. Anderson Cancer Center, Houston, Texas, USA and [3]Instituto de Tecnologia Química e Biológica, Universidade Nova de Lisboa, Lisbon, Portugal

Email: Romesh Stanislaus* - rstanisl@mdanderson.org; Mark Carey - mcarey@mdanderson.org; Helena F Deus - mhdeus@mdanderson.org; Kevin Coombes - kcoombes@mdanderson.org; Bryan T Hennessy - bhennessy@mdanderson.org; Gordon B Mills - gmills@mdanderson.org; Jonas S Almeida - jalmeida@mdanderson.org

* Corresponding author

Published: 22 December 2008

BMC Bioinformatics 2008, **9**:555 doi:10.1186/1471-2105-9-555

Received: 18 June 2008
Accepted: 22 December 2008

This article is available from: http://www.biomedcentral.com/1471-2105/9/555

Abstract

Background: Reverse Phase Protein Arrays (RPPA) are convenient assay platforms to investigate the presence of biomarkers in tissue lysates. As with other high-throughput technologies, substantial amounts of analytical data are generated. Over 1000 samples may be printed on a single nitrocellulose slide. Up to 100 different proteins may be assessed using immunoperoxidase or immunoflorescence techniques in order to determine relative amounts of protein expression in the samples of interest.

Results: In this report an RPPA Information Management System (RIMS) is described and made available with open source software. In order to implement the proposed system, we propose a metadata format known as reverse phase protein array markup language (RPPAML). RPPAML would enable researchers to describe, document and disseminate RPPA data. The complexity of the data structure needed to describe the results and the graphic tools necessary to visualize them require a software deployment distributed between a client and a server application. This was achieved without sacrificing interoperability between individual deployments through the use of an open source semantic database, S3DB. This data service backbone is available to multiple client side applications that can also access other server side deployments. The RIMS platform was designed to interoperate with other data analysis and data visualization tools such as Cytoscape.

Conclusion: The proposed RPPAML data format hopes to standardize RPPA data. Standardization of data would result in diverse client applications being able to operate on the same set of data. Additionally, having data in a standard format would enable data dissemination and data analysis.

Background

Reverse Phase Protein Arrays (RPPA) provide an analytical platform with the potential to characterize proteomic pathways similarly to the use of microarrays for gene expression studies. RPPAs are a high throughput tool for probing cell or tissue lysates that quantifies levels of selected proteins for which high quality antibody exist [1]. Consequently, RPPA analysis has the potential to be a

major tool in the high throughput screening of biopsies for markers of prognosis and therapy response in cancer and other complex diseases.

Protein microarrays can be classified into two groups: forward phase protein arrays (FPPA), and reverse phase protein arrays (RPPA). Forward phase protein arrays, also known as antibody arrays, employ high affinity bait molecules such as antibodies immobilized onto coated glass slides [2,3]. In reverse phase arrays, protein samples are immobilized on the slides and antibodies are used to probe the sample slides [4,5]. As a result of the differences in the immobilized medium in FPPA, one sample is probed against an array of antibodies, while in RPPA one antibody is arrayed against many samples. A major reason for its adoption could be its relative ease of producing high quality slides and its ability to quantify the amount of protein in the sample [4,6]. The ability to probe and quantify multiple samples for the expression of specific proteins in a single slide makes RPPA technology a good candidate for a high throughput analysis platform in a clinical setting.

Several analysis methods have been developed and used for quantifying signals in reverse phase protein arrays [1,5,7,8]. However, due to the large amounts and different types of data files resulting from RPPA experiments, there was a need to create an integrated platform for the management of data and integration of the available software. An integrated platform for RPPA data management becomes extremely important to organize and protect the data generated by a single experiment, and in particular helps organize both data and documentation for quality assurance purposes. Thus, at the core of the RPPA data management module is the data format known as the reverse phase protein array markup language (RPPAML). RPPAML and the data management module form the basis of an RPPA information management system (RIMS).

Consequently, three critical features are found particularly desirable in an RPPA information management system (RIMS). Firstly, the graphic visualization of the data must facilitate results reporting with specific reference to the array layout. Secondly, the data needs to be rendered in extensible markup language (XML) format (RPPAML) in order to make it easily portable to other applications and other information management systems. Thirdly, the analysis of results should include its visualization as biological networks, ideally using Cytoscape [9]. These three features outline an information system for RPPA data management that integrates processes and documents the entire experimental process. Seamless data integration and management are important success factors in proteomics experimentation and often its most time consuming [10]. RIMS hopes to provide a client side application, a repository to store the data and communication protocol based on XML for the storage and transport of the data.

Results

RIMS is an integrated platform for reverse phase protein array data management that consists of a client side application, central repository and a communication protocol based on RPPAML data structure. The client side application consists of an upload module and a visualization module.

Upload module

Data uploading and annotation is done using a separate module accessed through RPPA Data Manager GUI (please see Figure 1 for an overview of the process). Data imported is assembled with the annotation data into XML format, RPPAML (see under RPPAML metadata format for more information). The user can upload the data by pointing the application to the relevant folder and following the intuitive graphical user interface instruction. The user will be prompted to annotate the data once it has been pre-processed. Once the data is annotated, it can be saved on the local disk or uploaded to a web repository based on simple sloppy semantic database (S3DB) infrastructure [11].

RIMS is a client side application that interacts with the knowledge database (S3DB) to create a management infrastructure for RPPA data. RIMS interaction with S3DB is fully automated and location of S3DB can be distributed as long as it can be reached with a URL. Additionally, entry creation and data download is managed by RIMS software and there is no limit to the amount of data that can be uploaded or downloaded. RPPA Data Manager application manages the Upload and VisualizeGUI modules. Upload module lets the user annotate the data from what is gleaned by the software. The user enters the data through Excel templates, thereby eliminating the need to learn a new method (figure 1, box 1). Currently, the application supports MicroVigene® data for conversion to RPPAML. However, as new readers for RPPA analysis become available converters for these instruments will be made available.

Visualization module

RIMS provides many methods for data visualization, ranging from the scanned images of individual antibody arrays to the averages and standard deviations of individual samples on multiple arrays. VisualizeGUI module shows the integrated data in the context of sample names. It allows the user to export data in provided formats (RPPAML, Text, Excel, original format) for import into other applications. Additionally, the user can create correlation maps

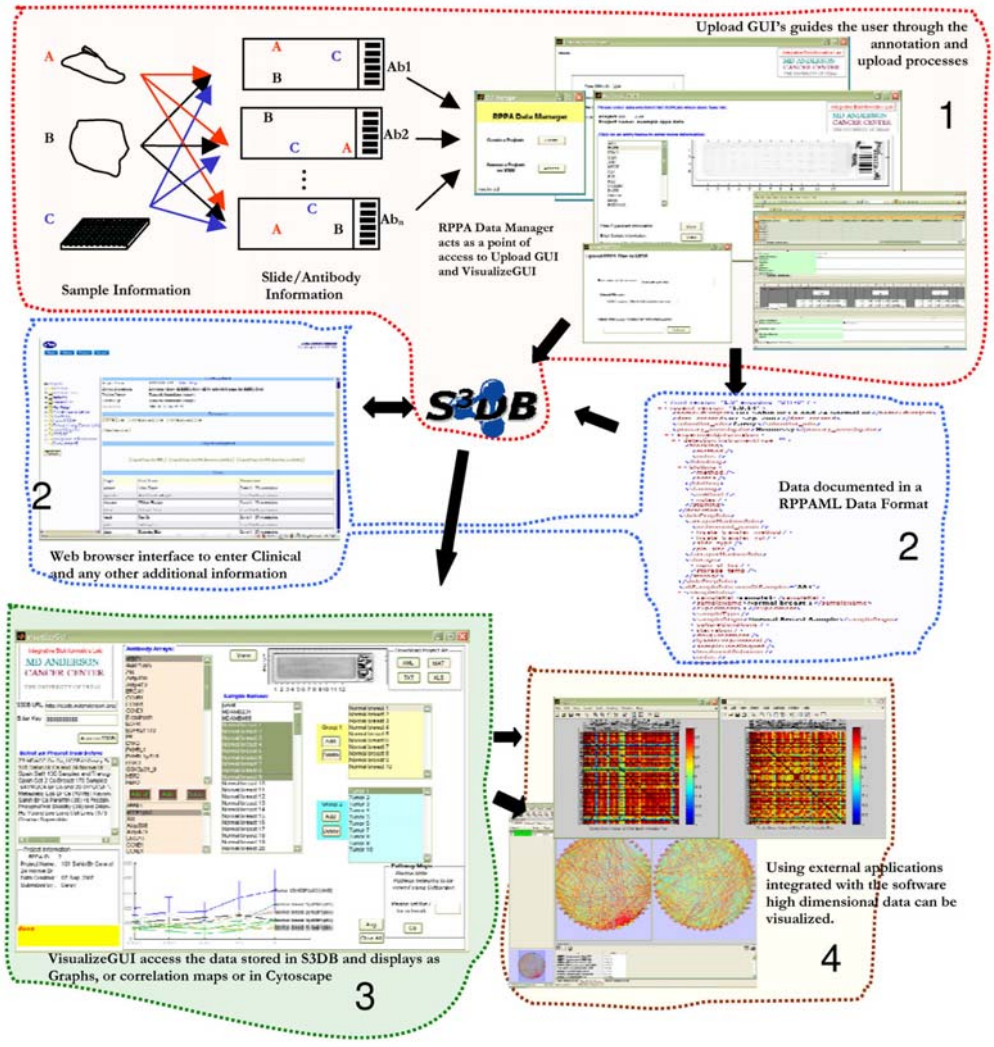


Figure 1

Flow of information from experimental samples to data visualization and analysis. The implementation architecture (shown in boxes in the figure) consists of a data management layer where data warehousing takes place (1). The next layer is the information and modeling layer where ontology and data standards can be applied to the data (2). Application integration layer provides the end user tools that interact with the data based on the ontology/standard (3). Dissemination layer (4) is used for knowledge delivery based on the data stored in the warehouse.

with the uploaded data. The data can also be exported to other applications for visualization (figure 1, box 3 & 4).

The RIMS client application also supports the creation of pathway maps for the selected antibody and sample lists. The generated pathways can be visualized using the popular Cytoscape tool [9] or through another application that supports the extensible graph markup and modeling language (XGMML) as an input format [12]. To generate pathway maps, the user can select corresponding groups (e.g. control (Group 1) Vs disease (Group 2)) and add them to the corresponding list boxes. Clicking the 'Go' button in the 'Pathway Maps' panel will generate the correlation maps from which pathways for the corresponding antibodies will be generated. Correlation maps and pathway maps will be displayed to the user and can then be saved or printed (Figure 1).

RPPAML metadata format

There are three options in the current version of the client application to export the data and processed results. The original data can be exported as a Microsoft Excel document, as a text file similar to the original upload format, as a Matlab® MAT file or as an XML document known as RPPAML. Exporting as an XML document is the most comprehensive option as it provides the original data with the context of its acquisition and processing, including the raw images. Since RPPAML is an application independent XML document, application developers using any programming language can access the data stored in the file. The RPPAML schema details can be found here: http://www.rppacentral.org/. A well formed RPPAML document contains minimum but sufficient information about the RPPA experiment and is defined as follows:

a) biological information: *sample biological information such as its provenance and treatment conditions, etc.*

b) antibody information: *validation information and approach*

c) detection information: *blocking, staining, amplification approach and antibody blotting information, etc.*

d) slide information: *slide preparation information such as array machine, lysate transfer method, pin or spot size, lysated amount, etc.*

e) data: *data about the experiment using the above conditions.*

Sub element <allSampleInfo> under the main element <experimentInformation> stores information about the biological sample such as its provenance, treatment conditions and other protocols (more details can be found on the website http://www.rppacentral.org under the schema tab). Also, under this main element sub-element <SlidePrepInfo> slide preparation information, such as array machine use and lysate transfer method used, is stored. Additionally, sub element <detection> stores information about blocking, staining and amplification approaches.

Element <arrays> describes all information about the reverse phase protein array (i.e. slide). Sub element <antibodyInfo> contains all information pertaining to an antibody used in the study. Additionally, <spotInfo> contains information about a spot in the slide and element <Img> contains the image of the slide as stored in any acceptable image file format. More information about individual elements is given on the website under the schema tab.

The schema describing the RPPAML structure is also represented in UML notation in Figure 2. This data model is the result of interaction between experimental researchers and bioinformaticians with the purpose of capturing all the relevant information for both data management and analysis. The proposed model documents the biological context, experimental conditions and data, thereby providing the data with the provenance and context and consequently preserving the granularity of the data set. Implementation of the model was achieved through the use of XML.

Federated repositories

RIMS uses S3DB as the data service backbone. The distributed nature of this component implies that individual users have the option of relying on locally installed S3DB deployments or using an external deployment such as the reference repository at The University of Texas M. D. Anderson Cancer Center [13]. As a consequence, individual users can access the data stored in these federated knowledge bases by simply pointing the application (RIMS) to any S3DB data service. A characteristic of S3DB semantic data services is that other data models describing complementary information can be integrated without compromising existing data [11]. This is particularly relevant for RPPA technology as new methods and improvements are devised for this young technology. However, using the proposed RPPAML data format, client applications will be aware of the context and provenance of data and provide the user with possible choices for analyzing the data.

Discussion

The convergence of information technology and biology has resulted in an unprecedented growth in the way researchers accumulate information. The main consequence of this growth can be seen in the realm of unprocessed data. Scientific data generated by experimental biologists has changed in scale, dimensionality and diversity. Gone are the days when data could be presented in a

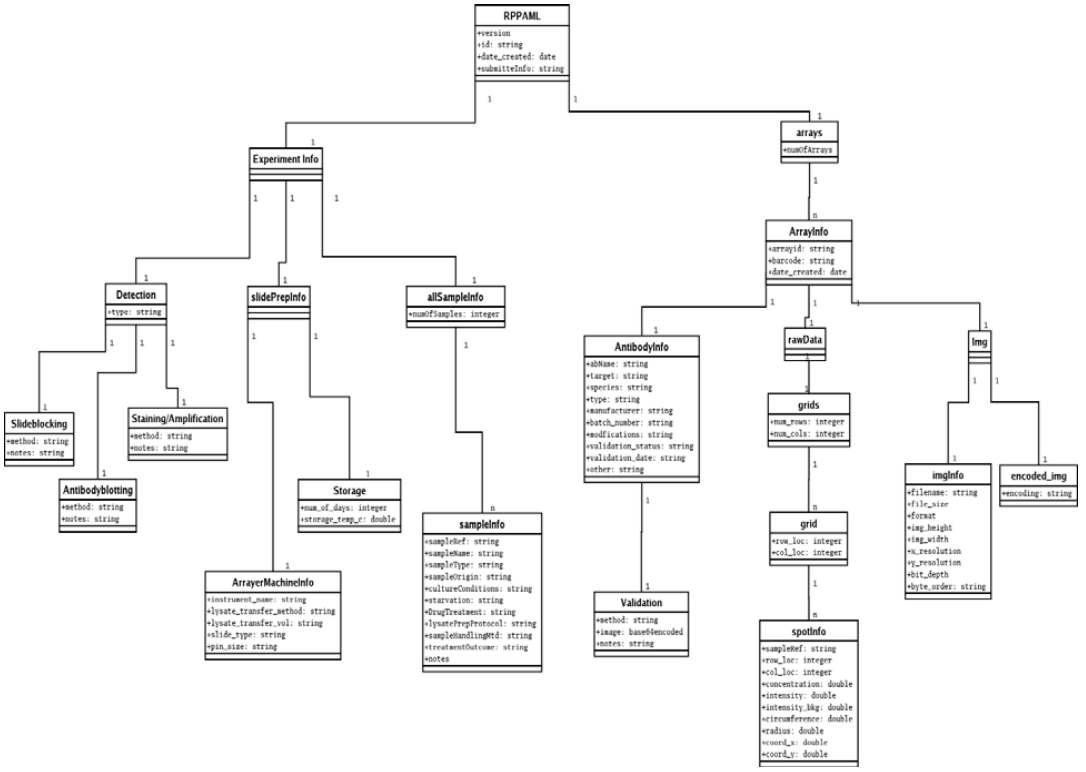


Figure 2
UML representation of the RPPAML data model. For more information on the RPPAML XML schema please go to: http://ibl.mdanderson.org/rppaml/schemaDocs/schema05_01312007.xsd

single or multiple spreadsheets. As data increase in diversity, scale and dimensionality within and between experiments, integrating data becomes a challenge. RIMS hopes to provide a single platform for raw, analysis and eventually clinical and other relevant data, and thereby provides the researcher with an integrated view of the data and analysis results for knowledge generation.

RIMS was created as a modular management platform for RPPA data. The modularity of design and the use of XML technologies will provide the ability to add new tools. For example, tools developed in the OOMPA toolkit http://bioinformatics.mdanderson.org/Software/OOMPA/ package can be ported and included in RIMS. This would allow the researcher to analyze the uploaded data seamlessly. The key feature of the proposed model is the RPPAML data format that would allow researchers to describe, document and disseminate RPPA data. Additionally, the structure format described here would enable development of third party software for the analysis of RPPA data. Another feature of RIMS is the client application described in this paper, which will manage data upload and retrieval of data from S3DB data services using the RPPAML data schema. This application is also fitted with import, preprocessing and graphic visualization and export tools. These tools a) process raw data files and images into the RPPA reference data structure; b) submit the data structure to an arbitrary S3DB data service (including creating the supporting data model in that service if it does not exist); c) include basic visualization tools such as rendering the original array image and aggregating results from multiple arrays by sample identity and concentration; d) export data in a variety of formats including a comprehensive RPPAML XML format and segmentation of cross-correlation tables into a network format that is rendered using Cytoscape. Both the client and source codes of RIMS are made freely available with an open source license [14].

Conclusion

The emergence of high throughput technologies also requires a standardized format to represent data generated from such processes. The proposed RPPAML schema provides RPPA high throughput technology with a standardized format to disseminate the data produced by such processes. Having such a format also enables different client applications to operate on the same set of data regardless of the instrument that produced the data. We hope that the proposed RPPAML format is the starting point in bringing vendors, analysts and scientists together to formulate community accepted standard for RPPA data. This in turn would enable this important technology to be widely available and useable.

Methods

RIMS stand alone client application uses MATLAB Component Runtime (MCR, no license needed by the user). This client application uses only client side computational resources and relies on a federated S3DB data services backbone http://www.s3db.org. Both components are available freely and with open source. The RIMS client is an integrated tool that guides the user from raw machine-generated data through annotation and analysis. Additionally, the RIMS tool allows the user to store the data in a federated S3DB database. The RIMS software module can be conveniently installed using Bioinformatic Station code distribution software [15]. This also allows the user to easily update/upgrade the RIMS software as new versions become available. RIMS provides tools for management, integration and analysis of RPPA data (Figure 1).

The implementation architecture (shown in colored boxes in Figure 1) consists of a data management layer where data warehousing takes place (Figure 1, box 1). The next layer is the information and modeling layer where ontology and data standards can be applied to the data (Figure 1, box 2). Information modeling is archived via the implementation of RPPAML through the use of XML. The application integration layer provides the end user with tools that interact with the data based on the ontology/standard (Figure 1, box 3). The dissemination layer (Figure 1, box 4) is used for knowledge delivery based on the data stored in the warehouse. This is done using graphs and integrating the output to external applications such as Cytoscape, etc.

Abbreviations

RPPA: reverse phase protein arrays; FPPA: forward phase protein arrays; XML: extensible markup language; RPPAML: reverse phase protein array markup language; RIMS: RPPA information management system; XGMML: eXtensible Graph Markup and Modeling Language; MCR: MATLAB Component Runtime; S3DB: simple sloppy semantic database

Availability and requirements

Project Name: http://www.rppacentral.org

Operating Systems: Windows XP

Programming Language: MATLAB

License: Open source project

Source Code: Available upon request. See website for contact details.

Authors' contributions

RS conceived the project and served as project lead. RS and JSA provided oversight of the project. HD managed data. MC and BH acquired and processed data. RS, JSA, KC and GBM prepared the manuscript. All authors read and approved the manuscript.

Acknowledgements

This work was supported in part by the National Heart, Lung and Blood Institute (NHLBI, N01-HV-28181), by the National Cancer Institute (NCI, P50 CA70907) of the US National Institutes of Health (NIH), by the Center for Clinical and Translational Sciences (1ULIRR024148), the Kleberg Center for Molecular Markers at M. D. Anderson Cancer Center, by NCI PO1CA099031, by The Susan G. Komen Foundation Biomarkers Identification and Validation Award FAS0703849, and U24 CA126479 and U24 CA126477. Authors would also like to thank Becky Partida for editing the manuscript.

References

1. Tibes R, Qiu Y, Lu Y, Hennessy B, Andreeff M, Mills GB, Kornblau SM: **Reverse phase protein array: validation of a novel proteomic technology and utility for analysis of primary leukemia specimens and hematopoietic stem cells.** *Mol Cancer Ther* 2006, **5(10):**2512-2521.
2. Haab BB: **Methods and applications of antibody microarrays in cancer research.** *Proteomics* 2003, **3(11):**2116-2122.
3. Nielsen UB, Geierstanger BH: **Multiplexed sandwich assays in microarray format.** *J Immunol Methods* 2004, **290(1–2):**107-120.
4. Charboneau L, Scott H, Chen T, Winters M, Petricoin EF 3rd, Liotta LA, Paweletz CP: **Utility of reverse phase protein arrays: applications to signalling pathways and human body arrays.** *Brief Funct Genomic Proteomic* 2002, **1(3):**305-315.
5. Espina V, Mehta AI, Winters ME, Calvert V, Wulfkuhle J, Petricoin EF 3rd, Liotta LA: **Protein microarrays: molecular profiling technologies for clinical specimens.** *Proteomics* 2003, **3(11):**2091-2100.
6. Liotta LA, Espina V, Mehta AI, Calvert V, Rosenblatt K, Geho D, Munson PJ, Young L, Wulfkuhle J, Petricoin EF 3rd: **Protein microarrays: meeting analytical challenges for clinical applications.** *Cancer Cell* 2003, **3(4):**317-325.
7. Grubb RL, Calvert VS, Wulkuhle JD, Paweletz CP, Linehan WM, Phillips JL, Chuaqui R, Valasco A, Gillespie J, Emmert-Buck M, Liotta LA, Petricoin EF: **Signal pathway profiling of prostate cancer using reverse phase protein arrays.** *Proteomics* 2003, **3(11):**2142-2146.
8. Hu J, He X, Baggerly KA, Coombes KR, Hennessy BT, Mills GB: **Non-parametric quantification of protein lysate arrays.** *Bioinformatics* 2007, **23(15):**1986-1994.
9. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T: **Cytoscape: a software environment for integrated models of biomolecular interaction networks.** *Genome Res* 2003, **13(11):**2498-2504.
10. Deutsch EW, Lam H, Aebersold R: **Data analysis and bioinformatics tools for tandem mass spectrometry in proteomics.** *Physiol Genomics* 2008, **33(1):**18-25.

11. Almeida JS, Chen C, Gorlitsky R, Stanislaus R, Aires-de-Sousa M, Eleuterio P, Carrico J, Maretzek A, Bohn A, Chang A, Zhang F, Mitra R, Mills GB, Wang X, Deus HF: **Data integration gets 'Sloppy'.** *Nat Biotechnol* 2006, **24(9):**1070-1071.
12. **XGMML** [http://www.cs.rpi.edu/~puninj/XGMML/]
13. **IBL** [http://ibl.mdanderson.org/s3db]
14. **RIMS** [http://www.rppacentral.org/]
15. **BioinformaticStation** [http://www.bioinformaticstation.org/]

OPEN ACCESS Freely available online

PLoS one

Exploratory Analysis of the Copy Number Alterations in Glioblastoma Multiforme

Pablo Freire[1,6], Marco Vilela[1,7], Helena Deus[1,7], Yong-Wan Kim[2], Dimpy Koul[2], Howard Colman[2], Kenneth D. Aldape[3], Oliver Bogler[4], W. K. Alfred Yung[2], Kevin Coombes[1], Gordon B. Mills[5], Ana T. Vasconcelos[6], Jonas S. Almeida[1]*

1 Department of Bioinformatics and Computational Biology, The University of Texas M. D. Anderson Cancer Center, Houston, Texas United States of America, **2** Department of Neuro-Oncology, The University of Texas M. D. Anderson Cancer Center, Houston, Texas United States of America, **3** Department of Pathology, The University of Texas M. D. Anderson Cancer Center, Houston, Texas United States of America, **4** Department of Neurosurgery, The University of Texas M. D. Anderson Cancer Center, Houston, Texas United States of America, **5** Department of Systems Biology, The University of Texas M. D. Anderson Cancer Center, Houston, Texas, United States of America, **6** Laboratório Nacional de Computação Científica, Laboratório de Bioinformática, Petrópolis, Rio de Janeiro, Brasil, **7** Instituto de Tecnologia Química e Biológica, Universidade Nova de Lisboa, Lisboa, Portugal

Abstract

Background: The Cancer Genome Atlas project (TCGA) has initiated the analysis of multiple samples of a variety of tumor types, starting with glioblastoma multiforme. The analytical methods encompass genomic and transcriptomic information, as well as demographic and clinical data about the sample donors. The data create the opportunity for a systematic screening of the components of the molecular machinery for features that may be associated with tumor formation. The wealth of existing mechanistic information about cancer cell biology provides a natural reference for the exploratory exercise.

Methodology/Principal Findings: Glioblastoma multiforme DNA copy number data was generated by The Cancer Genome Atlas project for 167 patients using 227 aCGH experiments, and was analyzed to build a catalog of aberrant regions. Genome screening was performed using an information theory approach in order to quantify aberration as a deviation from a centrality without the bias of untested assumptions about its parametric nature. A novel Cancer Genome Browser software application was developed and is made public to provide a user-friendly graphical interface in which the reported results can be reproduced. The application source code and stand alone executable are available at http://code.google.com/p/cancergenome and http://bioinformaticstation.org, respectively.

Conclusions/Significance: The most important known copy number alterations for glioblastoma were correctly recovered using entropy as a measure of aberration. Additional alterations were identified in different pathways, such as cell proliferation, cell junctions and neural development. Moreover, novel candidates for oncogenes and tumor suppressors were also detected. A detailed map of aberrant regions is provided.

Citation: Freire P, Vilela M, Deus H, Kim Y-W, Koul D, et al. (2008) Exploratory Analysis of the Copy Number Alterations in Glioblastoma Multiforme. PLoS ONE 3(12): e4076. doi:10.1371/journal.pone.0004076

Editor: Andrea Califano, Columbia University, United States of America

Received April 2, 2008; **Accepted** November 18, 2008; **Published** December 30, 2008

Funding: The authors thankfully acknowledge National Institutes of Health (NIH) support through the Center for Clinical and Translational Sciences under contract no. 1UL1RR024148 through awards RO1CA123304, and RO1CA056041. P Freire and AT Vasconcelos also acknowledge funding from the Brazilian government programs and Institutions CAPES, CNPq and FAPERJ.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: jalmeida@mdanderson.org

Introduction

Copy number alterations (CNAs) are known to be among the triggers of tumor formation [1,2]. Furthermore, tumor progression is associated with further variation in the copy number [3]. Although the association between chromosomal aberration and cancer has been known for a long time [4], recent advances in array-based techniques allowed a more refined description of the genomic structure, thus yielding a better characterization of copy number events.

Beginning with BAC and cDNA arrays [5,6], the resolution of the array techniques increased to the current sensitivity level capable of detecting events with a size of thousands of base pairs [7,8]. As a consequence of ongoing methodological advancement, a growing number of new oncogenes and tumor suppressors have been identified [9].

During cancer progression, tumor cells undergo several genomic changes. Mutations that enhance tumor progression are most likely to be positive selected in the neoplasm environment, and the cells that carry these mutations tend to be dominant in the tumor [10]. Due to the nature of CNAs, some mutations might carry genes that confer selective advantage, along with genes that do not. This creates a mutation background that obfuscates the localization of the major players of cancer [11]. Furthermore, tumor progression may take various routes as it occurs in the context of an individual genome and individual cell physiology. To track this variability, the analysis of several patients can be used to identify the recurrent regions of aberration (RRA).

Currently, most of the available mathematical tools for analyzing copy number data deal with segmentation methods and breakpoint detection [12,13]. These techniques are used to define discrete regions in the genome that have the same copy number, analyzing each sample individually. However, few studies have addressed the detection of RRA, which are common amplification or deletions that occur in the same locus in a group of samples. One common approach to detecting RRA is to define arbitrary thresholds to identify amplifications and deletions [14], using the frequency of events as a measure of abnormality for a given region in the genome. However, the signal from each experimental platform may differ in variance, which implies that a different threshold may be needed for each new analysis. Furthermore, tumor samples typically contain normal cells that contaminate the tumor DNA, thus altering the amplitude of copy number aberrations. Using absolute copy number values as thresholds to segment CNAs ignores both confounding effects. A number of other studies report more sophisticated methodologies for RRA detection, but they still rely on arbitrary calling for amplifications and deletions [11,15–17].

In this study, we propose a new method for identifying RRA based on the information content of each probe position. The main goal is to provide an approach that detects aberrant regions while making minimal assumptions about their nature, scale or prevalence.

Another aim of this study is to provide an exploratory framework for analyzing the data from glioblastoma multiforme (GBM) patients generated by The Cancer Genome Atlas project (TCGA;[18]). Despite the recent advances in the molecular pathology of GBM, the underlying mechanism of the origin and invasiveness of malignant glioma remains obscure [19].

As often noted in quality analysis surveys [20], data analysis results without dissemination of the applications that generated them are of unknown reproducibility. Therefore, an accompanying graphical tool is included to provide analysis of the copy number results as they are made available by the TCGA project and, more specifically, to allow the reproduction of the results reported here.

Results and Discussion

Exploring the TCGA data

In order to analyze the data generated by the TCGA project, a new graphical tool was developed, the Cancer Genome Browser (CGB), which is freely available at http://code.google.com/p/cancergenome/. The rationale for this tool is to provide a client application that can be used for the visualization and data processing reported here. In the tool, the input data is directly accessed from a semantic database [21] that provides the TCGA raw data in data structures designed to support the graphic representations reported here. The raw copy number data is stored alongside its preprocessed segmented values.

Assessment of aberration

We present a new mathematical method that uses Shannon's entropy as a measure of genomic aberration. The entropy measures the deviation from the common state in a system. In the genomic context, the common state would be that all the samples have a copy number around 2. Any deviation from that state should be reflected in the entropy so that the more aberrant a region is, the lower the entropy.

The detection of aberrant regions by the proposed procedure was first assessed using a simulation study (see Methods). Goodness of classification was determined using the area under the receiver operating characteristic curve, with 1 indicating perfect recognition of all alterations and 0.5 indicating random classification of the variation. Several simulated datasets were tested, encompassing different combinations of amplitudes and prevalences (the frequency of mutation in the population). In each one, a determined region of the genome had its copy number values added (amplified) by a certain amplitude value in a fraction of samples (controlled by the prevalence). The results were virtually the same when deletions were tested instead of amplifications (data not shown). The area under curve results are displayed in Figure 1, where it can be observed that, for amplitudes greater than 0.2, a perfect classification can be obtained if the prevalence is greater than 5%.

There are two main forms of CNA [11]: broad events, which can encompass several Mb or even the whole chromosome, and focal events, which are normally restricted to a few Mb. The search for new oncogenes and tumor suppressors in broad events can be extremely difficult due to the large number of genes within these regions. Therefore, we chose to analyze only the focal events and remove the influence of broad events in the entropy analysis. This was done by performing the analysis in each chromosome separately (thus nullifying the influence of whole chromosome aberrations) and applying baseline removal techniques to reduce the effects of other broad events in the entropy signal (see Methods).

GBM analysis

A total of 169 RRA were found using the 167 tumor samples (Table S1), being the majority of these regions annotated as copy number variation (CNV) that occurs in normal samples. CNV in normal cells has recently been described as a relatively common occurrence in the human genome [22]. To separate the mutations related to tumor progression from the normal CNV, the results were screened to identify the regions in which more than 50% of the probes were annotated as normal CNV or were detected in low-entropy peaks in normal samples. Thirty-one regions passed this test and were the objects of further analysis (Table 1 and Figure 2). The chromosomes X and Y were not analyzed. The entropy analyses for each chromosome are available on Figures S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19, S20, S21, S22.

Among those 31 regions, there are 10 genes known to modulate cell proliferation in GBM: EGFR, MDM2, MDM4, CDK4, PTEN, PDFGRA, CDKN2A, CDKN2C, NF1 and CHD5 [23,24]. However, the total number of CNAs involved in cell proliferation is still unknown, and recent studies have added new genes to the pool of oncogenes and suppressors of GBM [11,24,25].

Amplitude and prevalence of the CNA

The rationale of the entropy method allows a straightforward interplay between the prevalence and amplitude of CNAs. Some of the detected mutations, such as MDM4 and PTEN, have a low prevalence in the population but high copy number amplitude. Conversely, other regions, such as #51, have relatively low amplitude and a high prevalence.

Measures of the amplitude and prevalence directly on the log2 ratio copy number can lead to bias due to the whole chromosome variation and broad events. To avoid this problem, the amplification prevalence in an aberrant region was measured as the proportion of probes within the region, considering all samples, with a copy number above the 0.975 quantile of all copy number values for all samples at the same chromosome. For deletions, the proportion of values below the 0.025 quantile was used.

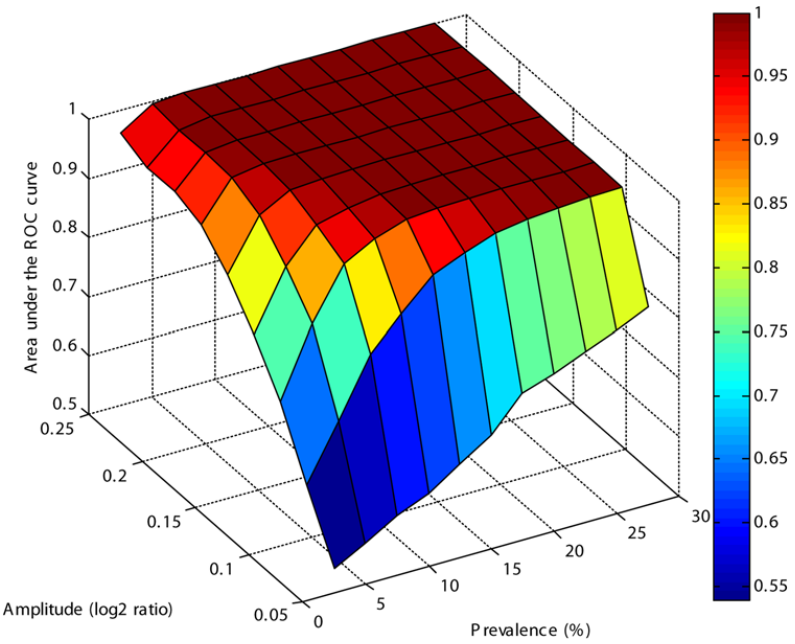


Figure 1. Evaluation of the entropy method performance by the area under the ROC curve from simulated datasets, accessing different amplitudes and prevalences of copy number aberrations. An area under the ROC curve of 1 means a perfect separation between mutated and normal regions, while a value of 0.5 means a random classification.
doi:10.1371/journal.pone.0004076.g001

The amplitude of a given probe position was obtained in a similar way: considering all samples and all probes within the region, the 0.975 and 0.025 quantiles of the copy number of the aberrant region were obtained. Then the quantile of these values, in comparison with the copy number for all the probes in the same chromosome, was used as a measure of amplitude called QQ0.025 and QQ0.975. For instance, let q be the 0.975 quantile among all the copy number values from all samples within a determined aberrant region. The amplitude of amplification, QQ0.975, for that region will be the quantile relative to q when compared to all of the copy number values for the same chromosome. If the observed region was not aberrant, QQ0.975 would be around 0.975. Consequently, a QQ0.975 close to 1 indicates amplification, and a QQ0.025 close to 0 indicates deletion. To consider an aberration as an amplification or deletion alike, the one tailed area is considered (which is to say, the area at either end of the two tailed distribution) and the amplification amplitude is expressed as 1- QQ0.975. Interestingly, the known aberrations for GBM tend to have the most extreme values of amplitude and are therefore found at the end of the distribution tails with QQ0.025 and 1-QQ0.975 values close to 0 (Figure 3). The separation between amplification and deletion is done by observing each value, QQ0.025 or 1-QQ0.975, is close to 0.

Genes within low-entropy regions

The recurrence of an aberration can be related to the influence of its genes in tumor progression [10]. However, some regions can contain other genes that may not be related to cancer. In [11], the genes that have influence on cancer are designated as "drivers" and the genes that are near the "drivers" but have no effect on tumor progression are designated as "passengers." Most authors assign only one "driver" per region [7,11,14], but there may be other genes related to cancer within the same region. For instance, region # 100 in Table 1, which contains the known described tumor suppressor CDKN2A, also contains the gene ELAVL2, which is related to neuronal proliferation and differentiation [26]. A cluster of interferon genes is also present in this region and can influence tumor growth and progression [27]. Another example of a region with more than one "driver" is region #117. Besides the well-described oncogene CDK4, this region also contains glioma-associated gene 1, which has been described as affecting cell proliferation and differentiation [28]. However, the CNA that is most likely to have multiple "drivers" is the well-known deletion of 1p36 [25,29]. This deletion is present in gliomas and neuroblastomas, but a single "driver" could not be defined [29] in previous studies. Together with the tumor suppressor CDH5, analysis of the low-entropy regions shows that the genes TNFRSF9, CAMTA1 and AJAP1 are among the candidates for tumor suppressors. A complete list of genes found in low-entropy regions is available in Table S2.

The gene CDKN2C, a well-known tumor suppressor [30], was found in region #9 (Table 1). Being an important player in oligodrendroglioma and medulloblastoma, the deletion of CDKN2C was recently described in GBM [17]. Moreover, a deletion of gene NF1, a gene associated with neurofibromatosis type 1 that appears to be a negative regulator of the Ras pathway [31], was also detected.

Among the candidates for suppressors in GBM listed in Table 1 are the genes LSAMP and ACCN1. The former has been described as a tumor suppressor in renal carcinoma [32], and the

Table 1. Amplifications and deletions in the GBM data.

Type	Region #	Known genes in GBM	Candidates	Ch	Start	End	Entropy	# of genes	Amplitude	Prevalence
Amplification	82	EGFR		7	54027966	55983910	−2.5865	7	0.000275	0.234739
	55	PDFGRA		4	53634367	57090231	−0.94003	22	0.000708	0.139728
	117	CDK4		12	56097914	56851736	−0.69274	26	0.001321	0.122698
	118	MDM2		12	67315903	67981156	−0.49955	7	0.000446	0.103178
	79			7	32978614	32991778	−0.48806	1	0.002777	0.101796
	22	MDM4		1	201942811	203355851	−0.48298	19	0.000311	0.084066
	34			2	113106316	113112980	−0.4263	0	0.000597	0.083832
	38			2	202864197	203906650	−0.26864	9	0.014994	0.038269
	51			3	181382003	181447344	−0.22919	0	0.000371	0.071856
	44			3	106776838	106776898	−0.21799	1	0.003143	0.047904
	91			7	152135252	152147233	−0.20795	1	0.005562	0.05988
	96		SGK3*	8	67783070	68102356	−0.18707	3	0.017116	0.041916
	27		NCOA1*	2	24616100	24712348	−0.18293	1	0.008564	0.053892
	26			2	24426149	24551934	−0.16724	1	0.008564	0.05988
	31		ATOH8*	2	85806436	85899917	−0.15732	1	0.012083	0.047904
	28			2	24819254	24866786	−0.15088	2	0.008564	0.047904
Deletion	100	CDKN2A		9	20336123	24769734	−1.89499	25	0.001192	0.258973
	112			11	70513145	70559392	−0.67419	0	0.018841	0.052181
	3	CHD5	AJAP1	1	4028404	6333694	−0.36827	10	0.004095	0.137835
	9	CDKN2C		1	50961735	51283220	−0.33261	2	0.000366	0.066068
	99			9	20240063	20280240	−0.31992	0	0.002421	0.147705
	83			7	86779323	86785351	−0.26366	0	0.000305	0.065868
	106	PTEN		10	89545841	89680005	−0.24913	3	0.000401	0.06373
	5		TNFRSF9	1	7889543	7926197	−0.23605	1	0.000965	0.155689
	2			1	3928417	3978407	−0.21835	0	0.006145	0.113772
	4		CAMTA1*	1	7728761	7742417	−0.21773	1	0.005078	0.149701
	146			17	28797037	29337463	−0.2158	1	0.01771	0.054943
	45		LSAMP	3	117520637	117525922	−0.2012	1	0.015102	0.05988
	145	NF1		17	26438606	26453055	−0.19121	1	0.004396	0.053892
	67			5	164370828	164522125	−0.19096	0	0.013057	0.05988
	147		ACCN1	17	29404279	29546036	−0.17745	1	0.021171	0.053892

Aberrant regions determined by the entropy method. These tables do not include regions for which more then half of the DNA probes were annotated as normal CNV. Candidate oncogenes / suppressors were selected by their potential association with cancer based on their molecular function or for being the only gene in the region. The genomic mapping was based on Build 18. The calling for amplifications and deletions was made by observing which amplitude measure, QQ0.025 (deletion) or 1-QQ0.975 (amplification), was closer to 0.

*Not described or validated as oncogene or tumor suppressor in previous studies.

doi:10.1371/journal.pone.0004076.t001

latter has been described as an inhibitor of glioma cell proliferation [33]. Some new candidates for oncogenes in GBM have also been found. The gene ATOH8 is a transcriptional regulator related to glial determination [34], but has never been described as an oncogene. Finally, non-annotated normal CNVs might be the cause of some low-entropy peaks, as in regions #91 and 112, which are located close to known CNV events.

Paralog regions may contribute to the pool of detected of aberrant regions. An example of the former is region #79, which is paralog to the region 55715461–55763010 on chromosome 7 [35],that lies within the EGFR amplified area, The Pearson's correlation between the copy number values of the two paralog regions is 0.79.

Comparison with other methods

The literature reports two methods for the identification of RRA that were applied to the TCGA glioblastoma dataset by the TCGA Research Network [36]: GISTIC [11] and GTS [17]. GISTIC uses an arbitrary threshold to define deletions and amplifications and calculates the q-value [37], which is an upper bound to the false discovery rate, as a measure of aberration. GTS searches for RRA using a statistic that considers the number of genes in aberrant area and their copy number value. It also uses arbitrary thresholds to define aberrant regions.

Despite the differences in the methods, the main mutations in GBM (EGFR, CDKN2A, CDKN2C, PDGFRA, PTEN, CDK4, MDM2 and MDM4) were correctly recovered by the entropy

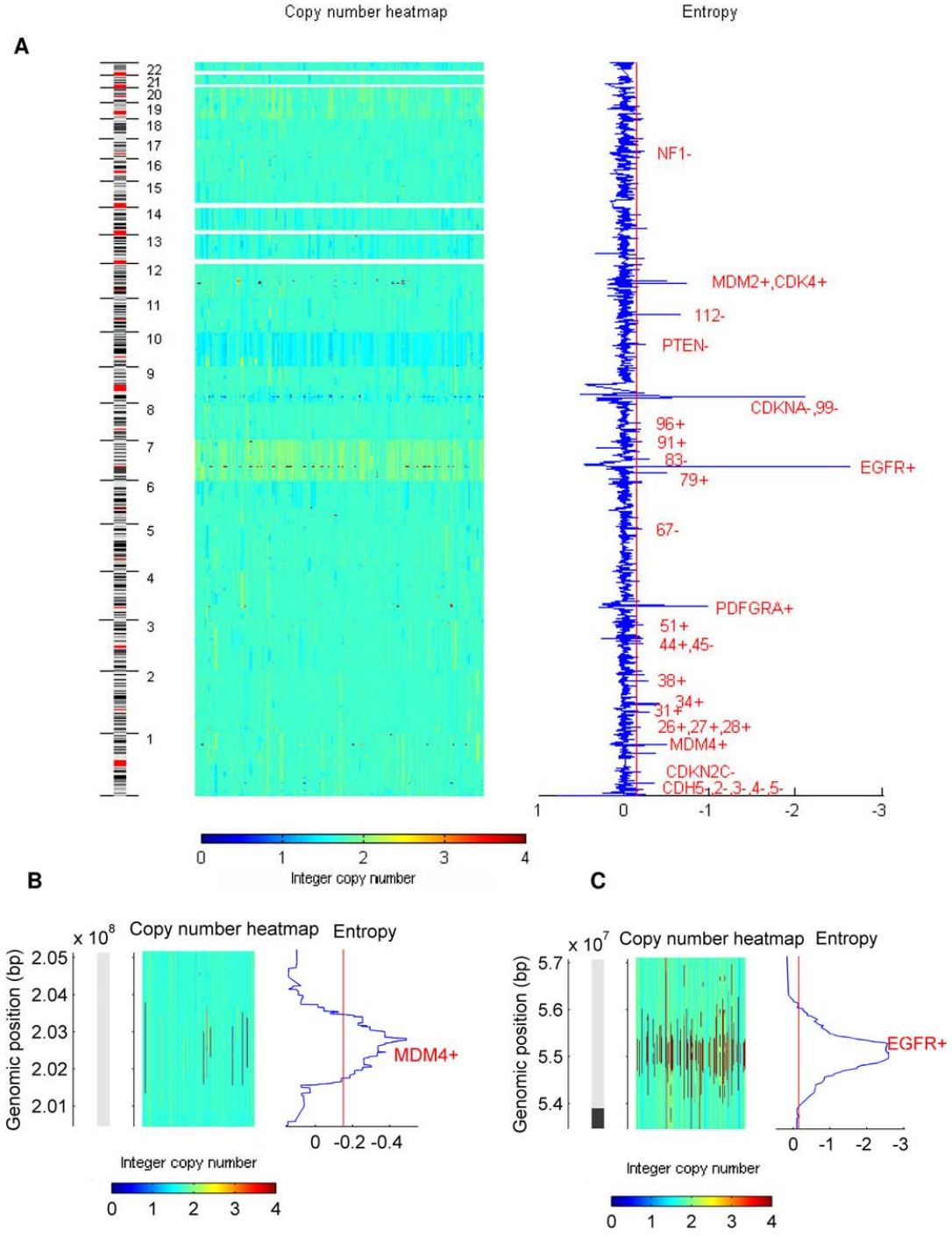

A
Copy number heatmap
Entropy
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
NF1-
MDM2+,CDK4+
112-
PTEN-
CDKNA-,99-
96+
91+
83-
EGFR+
79+
67-
PDFGRA+
51+
44+,45-
38+
34+
31+
26+,27+,28+
MDM4+
CDKN2C-
CDH5,2-,3-,4-,5-
1 0 -1 -2 -3
0 1 2 3 4
Integer copy number
B
x 10^8
Copy number heatmap
Entropy
Genomic position (bp)
2.05
2.04
2.03
2.02
2.01
MDM4+
0 -0.2 -0.4
Integer copy number
0 1 2 3 4
C
x 10^7
Copy number heatmap
Entropy
Genomic position (bp)
5.7
5.6
5.5
5.4
EGFR+
0 -1 -2 -3
Integer copy number
0 1 2 3 4

Figure 2. Entropy analysis for the GBM data entire genome (A), chromosome 1(B) and chromosome 7 (C). The segmented copy number values per sample are displayed as a heatmap on the left, with the tumor samples as columns; the entropy values are shown on the right. The gaps on the heatmap indicate genomic regions that lack coverage, such as the p arms of acrocentric chromosomes. The number in A corresponds to the region number in Table 1. Beside each region label, a plus sign indicates an amplification and a minus indicates a deletion. Note that a low-entropy region can be either a amplification or a deletion. The red line in the entropy plot shows the threshold for defining an aberrant region, which is the 0.05 quantile of the bootstrap distribution of the entropy. Peaks that are below the threshold but have no region assigned are normal CNV. The cytoband annotation was retrieved using the UCSC Table Browser.
doi:10.1371/journal.pone.0004076.g002

method and the TCGA Research Network analysis, which used a combination of methods that included GISTIC and GTS. Confirming the simulation results, the entropy analysis was insensitive to mutations with low prevalence (<4–5%). Some known oncogenes and suppressors in GBM were not detected by the entropy method, but were correctly identified by the combined GISTIC and GTS analysis (prevalence in parentheses): MET (3%), CDK6 (1%), TP53 (1%), CCND2 (2%) and PIK3CA (2%). The tumor suppressor RB1 was not detected by entropy because it is located in the peak of a broad deletion event and is obfuscated by the baseline removal. However, low prevalence mutations will always represent a challenge for statistical methods that consider the whole population in the analysis. With arbitrary thresholds for amplification and deletion of 1.5 and −1.5, respectively (log2 scale), some mutations were not detected by any method, such as CCNE1 and CCND3 (genes with a prevalence less than 1%).

The absence of any unique pattern on the GBM genotype and the influence of low prevalence mutations suggest that a better description of the copy number data can be achieved if individual characteristics of each sample is considered instead of a summary for the whole population. In that context, the entropy method should be used as an initial scan of the copy number data due to its speed, in the order of seconds, robustness and lack of parameter calibration. Future versions of the CGB tool will include GTS and GISTIC, thus allowing the integration of different methods of RRA detection with heatmap visualization and data exporting.

Tumor vs. normal samples

In this study, normal samples were used for identifying germline CNV (see Metods). However, the comparison of the paired normal and tumor samples reveals mutations that are only present in the normal samples, which contradicts the common assumption that blood samples contain only germline CNV (Figure 4). A detailed analysis of these mutations indicates that most of them are artifacts of the segmentation procedure, which creates very small segments in the normal samples that are not present in the tumor samples. Also, the samples from patient TCGA-06-0178 appears to be mislabeled; while the tumor sample has almost no mutation, the blood sample contains several CNAs, including the oncogene CDK4.

In a comparison between the low-entropy regions on the normal samples (Table S3) and the regions annotated in the CNV databases (described in Methods), 62% of the DNA probes of the low-entropy regions in normal samples are also located in known CNV regions. Moreover, some of the low-entropy regions are located close to a CNV, and it might be reasonable to assume that they are part of the CNV, once it is difficult to achieve a precise definition of the boundaries of a CNV with array techniques [38]. This observation suggests incompleteness of the current databases for normal CNV. Indeed, some of the low-entropy regions in normal samples (e.g. region 85219839–85227131 on chromosome 12) were later confirmed by sequencing to be CNVs [38]. Experimental artifacts may also be underlying reason for the low-entropy regions in normal samples.

Conclusion

This study presents a new method for detecting RRA that uses low entropy as an indicator. A stand-alone graphic application is provided for the exploration of the TCGA data and the replication of the detection of low-entropy regions presented here.

From a dataset of 167 GBM samples from the TCGA project, 31 aberrant regions were found, including 10 known CNAs in GBM, namely the genes EGFR, MDM2, MDM4, CDK4, PDGFRA, PTEN, CDKN2A, CDKN2C, NF1 and CHD5. Also, candidates that were never described as being major players in cancer, such as the glial differentiation gene ATOH8 and the transcription factor NCOA1, were detected in aberrant regions. The unusual level of enrichment of the list of candidate oncogenes and tumor suppressors lends considerable expectation to those few regions for which neither variability nor association with tumor formation could be found.

The analysis of the entropy in the blood (non-tumor) samples showed that only 62% of the aberrant regions were previously annotated as normal CNV regions. The expansion of the CNV databases may refine the separation between normal CNV and copy number aberrations that have influence on cancer.

Methods

Source data

A total of 227 normalized array comparative genomic hybridization (aCGH) results for GBM patients were retrieved from the TCGA data portal (http://tcga-data.nci.nih.gov/). The aCGH experiments were performed by the Memorial Sloan-Kettering Cancer Center using the Agilent Human Genome CGH Microarray 244A (Agilent Technologies, Inc., Santa Clara, California) platform. From the 227 samples (Table S4), 167 were tumor samples and 60 were blood samples. When there was more than one sample of the same tissue for a patient, one sample was randomly selected (see Supplementary Material for a sample list). Of the 167 tumor samples, 58 had a paired blood sample from the same patient.

The normalized copy number data obtained from the 227 samples were mapped into the human genome using the Build 18 (NCBI 36) assembly with an annotation file provided by the manufacturer (http://www.chem.agilent.com/). The array normalization procedure was performed by Memorial Sloan Kettering Cancer Center with their in–house algorithm that corrects for CG contents bias (see TCGA Data portal; http://tcga.cancer.gov/dataportal). The copy number data was filtered using the Circular Binary Segmentation (CBS) algorithm as implemented in the R package DNAcopy with the default parameter settings [12].

Data analysis method

The detection of aberrations was pursued here as that of an unqualified deviation. As a consequence of this critical concern with untested null models, the choice of method must satisfy two concerns about possible bias. First, it must make no assumptions about a reference non-deviant signal. Secondly, it must make no assumptions about the shape of the variation. These non-

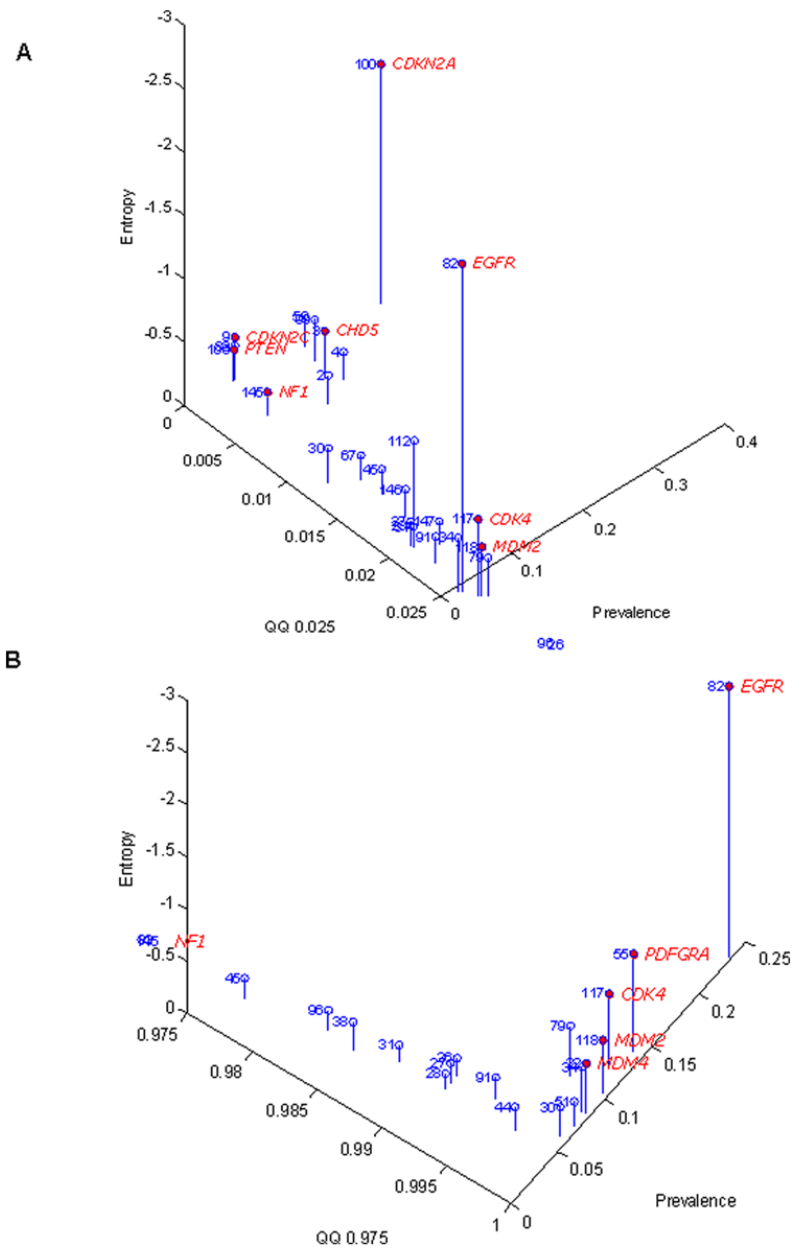


Figure 3. Interplay between the amplitude, prevalence and entropy in deletions (A) and amplifications (B). The prevalence measure adopted is the proportion of DNA probes on determined regions that have a log2 ratio above or below the 0.025 and 0.975 quantiles.
doi:10.1371/journal.pone.0004076.g003

parametric requirements are satisfied by approaches that use the density of observed measures to assess the information content of the signal. The individual signal is thus assessed by the probability, p, of the deviation in the context of observed signals. Shannon's entropy (Eq. 1) was calculated for each of the DNA probe positions, i = 1,…,n.

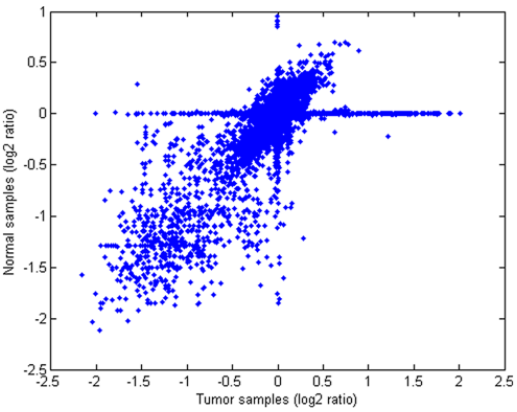


Figure 4. Log2 ratio of copy number of the 58 tumor-normal sample pairs. The values in the diagonal line correspond to variations similarly observed in tumor and normal tissue. The values in the horizontal line correspond to amplifications (right) and deletions (left) observed only in the tumor samples.
doi:10.1371/journal.pone.0004076.g004

$$S_i = -\sum_{j=1}^{m} p_{ij} \log_2(p_{ij}) \quad (1)$$

The probability, p_{ij}, for each copy number value was determined as the fraction of the kernel density, K, observed in all samples, $j = 1,\ldots m$, at that position (Eq. 2).

$$p_{ij} = K(CN_{i,j}) \Big/ \max_i \left(\sum_{j=1}^{m} K(CN_{ij}) \right) \quad (2)$$

The Parzen window method [39] with the Gaussian kernel function was used to approximate the probability density value K of the log2 ratio of copy numbers observed at that position, $CN_{i,j}$. This technique considers that every element in the population is a center of a Gaussian curve, and that the probability density value for a given point is the sum of all Gaussian values at that point. The calculation of the kernel density for all DNA probes would have required a large amount of computational effort. Therefore, the kernel density was sampled in 100 equally-distributed points, KS, ranging from the minimum to the maximum value of the copy number log2 ratio (Eq. 3). The probability density value, $K(CN_{ij})$, was then obtained by interpolation with the vector KS. The parameter σ, relative to the bandwidth of the kernel, was defined as the standard deviation of the raw data inside each segment summarized for all segments in all samples. Methods for bandwidth estimation that were designed for Gaussian populations yielded a bandwidth parameter that was too short, which resulted in several peaks in the probability density distribution (data not shown). Our bandwidth selection criteria resulted in a unimodal probability density centered at 0. Since most of the important CNAs have high amplitudes, and consequently low probably densities, the detection of aberrant regions is relatively insensitive to large bandwidth parameters.

$$K(KS_q) = \frac{1}{\sqrt{2\pi}\sigma} \sum_{i=1}^{m} \sum_{j=1}^{n} e^{-\frac{\left(KS_q - CN_{ij}\right)^2}{2\sigma^2}} \quad (3)$$

The amount of information associated with an "aberration" is inversely proportional to the entropy S. If a determined region is recurrently amplified or deleted, it should have a higher information content, and thus a lower entropy, when compared to the overall distribution of the entropy.

The implementation of this three step procedure is detailed using Matlab's m-code :

```
[n,m]=size(X);
```

1. Generate reference distribution

   ```
   [f,xi]=ksdensity(X(:), bandwidth);
   ```

   and replace each value by its density (Eq. 3)

   ```
   P=X;P(:)=interp1(xi,f,X(:));
   ```

2. Calculate the actual probability now as the proportion of row density (Eq. 2)

   ```
   P=P./repmat(max(sum(P,2)),1,m);
   ```

3. Calculate Shannon's entropy (Eq. 1)

   ```
   H=-sum(P.*log2(P),2);
   ```

Although applied to aCGH experiments in this work, the entropy method is suitable for any array-based copy number platform.

Detecting the regions of interest

As discussed in [11], there are two main forms of CNAs in tumor cells: broad events, which can contain several Mb of nucleotides and encompass numerous genes; and focal events, which are much more localized. Focal events inside broad events represent a challenge for methods that are based on thresholds for the binary calling of amplifications and deletions, once the entire broad region can be considered significant, hence hidden the focal events. However, some methods for RRA detection, while relying on arbitrary thresholds, use the amplitude to separate these nested focal events [11,17].

Even though broad events can be prevalent in the cancer genome [14], their applicability for finding new oncogenes or tumor suppressors is limited due to the large number of genes present in those regions. Thus, in this paper the detection of RRA was limited to the focal events. To remove the influence of entire chromosome amplification or deletion, the kernel density was calculated individually for each chromosome. Moreover, to diminish the effects of broad events on the entropy, the baseline of the entropy signal was removed using a Whitaker filter [40] (smoothing). For each probe position, the value of the entropy was determined as follows: Final entropy = Original entropy − Smoothed entropy. Therefore, only peaks in the entropy, which represent focal events, remained in the signal. Finally, a threshold for the entropy was obtained using the 0.05 quantile of the bootstrap distribution of the entropy. The regions that had a final entropy lower than the threshold were considered RRA. Regions represented by only one probe were not considered.

In the CGB tool, the baseline removal is given as an option to the user. Therefore, it is possible to deactivate this procedure in

order to analyze broad events as well. Since the entropy method does not consider the size of the events, it is capable of detecting broad events such arm-size or even whole chromosome events. For whole chromosome events, the entropy should be measured in the whole genome instead of individually on each chromosome.

Identifying normal CNV

CNV in normal cells has recently been described as a relatively common occurrence in the human genome [22]. To detect whether an RRA is a normal copy number variation or an aberrant alteration that promotes cell proliferation, the regions were compared to the entries of the Database of Genomic Variants (http://projects.tcag.ca/variation/; version 18v1; [22]) and the "Structural Variants" annotations in the UCSC Genome Browser [41]. Also, the entropy was calculated for the 60 normal samples using the same procedure described above. The low-entropy regions in normal samples were not used when analyzing the tumor dataset.

Simulation of aberrant regions

One hundred simulations were performed to analyze the behavior of the entropy according to variations in the amplitude and the prevalence of CNAs. The length of each aberration was not changed since our method considers each position independently.

A set with 100 artificial patients was built using randomly sampled copy number values from the GBM data. The simulated CNA amplitude ranged from 0 to 0.4 (log2 ratio scale) with a prevalence from 0 to 25%. The area under the receiver operator characteristic curve (ROC) was used for performance evaluation in each simulated condition. The analysis of the simulation is described in the Results section.

Supporting Information

Figure S1 Entropy analysis of chromosome 1, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples. Found at: doi:10.1371/journal.pone.0004076.s001 (0.26 MB TIF)

Figure S2 Entropy analysis of chromosome 2, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples. Found at: doi:10.1371/journal.pone.0004076.s002 (0.34 MB TIF)

Figure S3 Entropy analysis of chromosome 3, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples. Found at: doi:10.1371/journal.pone.0004076.s003 (0.33 MB TIF)

Figure S4 Entropy analysis of chromosome 4, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples. Found at: doi:10.1371/journal.pone.0004076.s004 (0.34 MB TIF)

Figure S5 Entropy analysis of chromosome 5, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples. Found at: doi:10.1371/journal.pone.0004076.s005 (0.32 MB TIF)

Figure S6 Entropy analysis of chromosome 6, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples. Found at: doi:10.1371/journal.pone.0004076.s006 (0.37 MB TIF)

Figure S7 Entropy analysis of chromosome 7, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples. Found at: doi:10.1371/journal.pone.0004076.s007 (0.48 MB TIF)

Figure S8 Entropy analysis of chromosome 8, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples. Found at: doi:10.1371/journal.pone.0004076.s008 (0.34 MB TIF)

Figure S9 Entropy analysis of chromosome 9, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples. Found at: doi:10.1371/journal.pone.0004076.s009 (0.35 MB TIF)

Figure S10 Entropy analysis of chromosome 10, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples. Found at: doi:10.1371/journal.pone.0004076.s010 (0.40 MB TIF)

Figure S11 Entropy analysis of chromosome 11, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples.
Found at: doi:10.1371/journal.pone.0004076.s011 (0.39 MB TIF)

Figure S12 Entropy analysis of chromosome 12, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples.
Found at: doi:10.1371/journal.pone.0004076.s012 (0.39 MB TIF)

Figure S13 Entropy analysis of chromosome 13, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples.
Found at: doi:10.1371/journal.pone.0004076.s013 (0.38 MB TIF)

Figure S14 Entropy analysis of chromosome 14, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples.
Found at: doi:10.1371/journal.pone.0004076.s014 (0.39 MB TIF)

Figure S15 Entropy analysis of chromosome 15, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples.
Found at: doi:10.1371/journal.pone.0004076.s015 (0.34 MB TIF)

Figure S16 Entropy analysis of chromosome 16, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples.
Found at: doi:10.1371/journal.pone.0004076.s016 (0.32 MB TIF)

Figure S17 Entropy analysis of chromosome 17, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples.
Found at: doi:10.1371/journal.pone.0004076.s017 (0.41 MB TIF)

Figure S18 Entropy analysis of chromosome 18, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples.
Found at: doi:10.1371/journal.pone.0004076.s018 (0.39 MB TIF)

Figure S19 Entropy analysis of chromosome 19, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples.
Found at: doi:10.1371/journal.pone.0004076.s019 (0.38 MB TIF)

Figure S20 Entropy analysis of chromosome 20, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples.
Found at: doi:10.1371/journal.pone.0004076.s020 (0.42 MB TIF)

Figure S21 Entropy analysis of chromosome 21, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples.
Found at: doi:10.1371/journal.pone.0004076.s021 (0.40 MB TIF)

Figure S22 Entropy analysis of chromosome 22, containing the copy number heatmap (on the right) and the entropy signal (left). The threshold for determining aberrant regions is displayed in the entropy plot as a red line, and it is defined by the quantile 0.05 of the bootstrap distribution of entropy. Only tumor samples are included. The assignments of the regions is the same on the Table 1 of the manuscript and peaks that don't have any regions assigned represent normal CNV or low-entropy regions in normal samples.
Found at: doi:10.1371/journal.pone.0004076.s022 (0.39 MB TIF)

Table S1 Low-entropy regions, including the normal CNV regions.
Found at: doi:10.1371/journal.pone.0004076.s023 (0.10 MB XLS)

Table S2 Genes within low-entropy regions
Found at: doi:10.1371/journal.pone.0004076.s024 (0.05 MB XLS)

Table S3 Low-entropy regions in the 60 normal samples.
Found at: doi:10.1371/journal.pone.0004076.s025 (0.22 MB XLS)

Table S4 List of the samples used and the tissues where it came from. A sample code is composed by two parts, the patient code

and the tissue code. For instance, the sample TCGA-02-0001-01A can be devided in tow parts: TCGA-02-0001 (patient code) and -01A (tissue coide). Therefor, the samples TCGA-02-0001-01A and TCGA-02-0001-10A came from the same patient. For further information on the sample barcode, please refer the TCGA data description (http://tcga-data.nci.nih.gov/docs/TCGA_Data_Primer.pdf).

Found at: doi:10.1371/journal.pone.0004076.s026 (0.03 MB XLS)

Acknowledgments

The authors thankfully acknowledge the TCGA research network for providing the data for this work, especially the Memorial Sloan-Kettering Cancer Center for performing the aCGH experiments and normalization. The authors also thank Rebecca Partida for reviewing the manuscript.

Author Contributions

Conceived and designed the experiments: PF WKAY KC GM ATV JSA. Performed the experiments: PF ATV. Analyzed the data: PF MV YWK DK HC KA OB WKAY ATV JSA. Contributed reagents/materials/analysis tools: PF HFD WKAY ATV JSA. Wrote the paper: PF ATV JSA.

References

1. Beckmann MW, Niederacher D, Schnurch HG, Gusterson BA, Bender HG (1997) Multistep carcinogenesis of breast cancer and tumour heterogeneity. J Mol Med 75: 429–439.
2. Weir B, Zhao X, Meyerson M (2004) Somatic alterations in the human cancer genome. Cancer Cell 6: 433–438.
3. Santos GC, Zielenska M, Prasad M, Squire JA (2007) Chromosome 6p amplification and cancer progression. J Clin Pathol 60: 1–7.
4. Bretland Farmer J, Moore JES, Walker CE (1905) On the Cytology of Malignant Grows. Proceedings of the Royal Society of London Series B, Containing Papers of a Biological Character 77: 336–353.
5. Pollack JR, Perou CM, Alizadeh AA, Eisen MB, Pergamenschikov A, et al. (1999) Genome-wide analysis of DNA copy-number changes using cDNA microarrays. Nat Genet 23: 41–46.
6. Pinkel D, Segraves R, Sudar D, Clark S, Poole I, et al. (1998) High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. Nat Genet 20: 207–211.
7. Brennan C, Zhang Y, Leo C, Feng B, Cauwels C, et al. (2004) High-resolution global profiling of genomic alterations with long oligonucleotide microarray. Cancer Res 64: 4744–4748.
8. Coe BP, Ylstra B, Carvalho B, Meijer GA, Macaulay C, et al. (2007) Resolving the resolution of array CGH. Genomics 89: 647–653.
9. Kallioniemi A (2007) CGH microarrays and cancer. Curr Opin Biotechnol.
10. Merlo LM, Pepper JW, Reid BJ, Maley CC (2006) Cancer as an evolutionary and ecological process. Nature Reviews Cancer 6: 924–935.
11. Beroukhim R, Getz G, Nghiemphu L, Barretina J, Hsueh T, et al. (2007) Assessing the significance of chromosomal aberrations in cancer: methodology and application to glioma. Proc Natl Acad Sci U S A 104: 20007–20012.
12. Venkatraman ES, Olshen AB (2007) A faster circular binary segmentation algorithm for the analysis of array CGH data. Bioinformatics 23: 657–663.
13. Picard F, Robin S, Lavielle M, Vaisse C, Daudin JJ (2005) A statistical approach for array CGH data analysis. BMC Bioinformatics 6: 27.
14. Maher EA, Brennan C, Wen PY, Durso L, Ligon KL, et al. (2006) Marked genomic differences characterize primary and secondary glioblastoma subtypes and identify two distinct molecular and clinical secondary glioblastoma entities. Cancer Res 66: 11502–11513.
15. Diskin SJ, Eck T, Greshock J, Mosse YP, Naylor T, et al. (2006) STAC: A method for testing the significance of DNA copy number aberrations across multiple array-CGH experiments. Genome Res 16: 1149–1158.
16. Guttman M, Mies C, Dudycz-Sulicz K, Diskin SJ, Baldwin DA, et al. (2007) Assessing the significance of conserved genomic aberrations using high resolution genomic microarrays. PLoS Genet 3: e143.
17. Wiedemeyer R, Brennan C, Heffernan TP, Xiao Y, Mahoney J, et al. (2008) Feedback circuit among INK4 tumor suppressors constrains human glioblastoma development. Cancer Cell 13: 355–364.
18. TCGA Research Network (2008) Comprehensive genomic characterization defines human glioblastoma genes and core pathways. Nature 455: 1061–1068.
19. Louis DN (2006) Molecular pathology of malignant gliomas. Annu Rev Pathol 1: 97–117.
20. Coombes KR, Wang J, Baggerly KA (2007) Microarrays: retracing steps. Nat Med 13: 1276–1277; author reply 1277–1278.
21. Almeida JS, Chen C, Gorlitsky R, Stanislaus R, Aires-de-Sousa M, et al. (2006) Data integration gets 'Sloppy'. Nat Biotechnol 24: 1070–1071.
22. Iafrate AJ, Feuk L, Rivera MN, Listewnik ML, Donahoe PK, et al. (2004) Detection of large-scale variation in the human genome. Nat Genet 36: 949–951.
23. Reifenberger G, Collins VP (2004) Pathology and molecular genetics of astrocytic gliomas. J Mol Med 82: 656–670.
24. Bagchi A, Papazoglu C, Wu Y, Capurso D, Brodt M, et al. (2007) CHD5 is a tumor suppressor at human 1p36. Cell 128: 459–475.
25. Ichimura K, Vogazianou AP, Liu L, Pearson DM, Backlund LM, et al. (2007) 1p36 is a preferential target of chromosome 1 deletions in astrocytic tumours and homozygously deleted in a subset of glioblastomas. Oncogene.
26. Yano M, Okano HJ, Okano H (2005) Involvement of Hu and heterogeneous nuclear ribonucleoprotein K in neuronal differentiation through p21 mRNA post-transcriptional regulation. J Biol Chem 280: 12690–12699.
27. Connett JM, Badri L, Giordano TJ, Connett WC, Doherty GM (2005) Interferon regulatory factor 1 (IRF-1) and IRF-2 expression in breast cancer tissue microarrays. J Interferon Cytokine Res 25: 587–594.
28. Di Marcotullio L, Ferretti E, Greco A, De Smaele E, Po A, et al. (2006) Numb is a suppressor of Hedgehog signalling and targets Gli1 for Itch-dependent ubiquitination. Nat Cell Biol 8: 1415–1423.
29. Okawa ER, Gotoh T, Manne J, Igarashi J, Fujita T, et al. (2008) Expression and sequence analysis of candidates for the 1p36.31 tumor suppressor gene deleted in neuroblastomas. Oncogene 27: 803–810.
30. Uziel T, Zindy F, Sherr CJ, Roussel MF (2006) The CDK inhibitor p18Ink4c is a tumor suppressor in medulloblastoma. Cell Cycle 5: 363–365.
31. Li F, Munchhof AM, White HA, Mead LE, Krier TR, et al. (2006) Neurofibromin is a novel regulator of RAS-induced signals in primary vascular smooth muscle cells. Hum Mol Genet 15: 1921–1930.
32. Chen J, Lui WO, Vos MD, Clark GJ, Takahashi M, et al. (2003) The t(1;3) breakpoint-spanning genes LSAMP and NORE1 are involved in clear cell renal cell carcinomas. Cancer Cell 4: 405–413.
33. Vandepoele K, Andries V, Van Roy N, Staes K, Vandesompele J, et al. (2008) A constitutional translocation t(1;17)(p36.2;q11.2) in a neuroblastoma patient disrupts the human NBPF1 and ACCN1 genes. PLoS ONE 3: e2207.
34. Inoue C, Bae SK, Takatsuka K, Inoue T, Bessho Y, et al. (2001) Math6, a bHLH gene expressed in the developing nervous system, regulates neuronal versus glial differentiation. Genes Cells 6: 977–986.
35. Bailey JA, Yavor AM, Massa HF, Trask BJ, Eichler EE (2001) Segmental duplications: organization and impact within the current human genome project assembly. Genome Res 11: 1005–1017.
36. McLendon R, Friedman A, Bigner D, Van Meir EG, Brat DJ, et al. (2008) Comprehensive genomic characterization defines human glioblastoma genes and core pathways. Nature.
37. Storey JD, Tibshirani R (2003) Statistical significance for genomewide studies. Proc Natl Acad Sci U S A 100: 9440–9445.
38. Kidd JM, Cooper GM, Donahue WF, Hayden HS, Sampas N, et al. (2008) Mapping and sequencing of structural variation from eight human genomes. Nature 453: 56–64.
39. Parzen E (1956) On Consistent Estimates of the Spectral Density of a Stationary Time Series. Proc Natl Acad Sci U S A 42: 154–157.
40. Eilers PH (2003) A perfect smoother. Anal Chem 75: 3631–3636.
41. Kuhn RM, Karolchik D, Zweig AS, Trumbower H, Thomas DJ, et al. (2007) The UCSC genome browser database: update 2007. Nucleic Acids Res 35: D668–673.

ITQB-UNL | Av. da República, 2780-157 Oeiras, Portugal
Tel (+351) 214 469 100
Fax (+351) 214 411 277

www.itqb.unl.pt